Design of Generic Enterprise Software:

Considerations for Cultivating a Knowledge Infrastructure

Jacob Kleivdal Ommundsen & Sigvart Bretteville-Jensen

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

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

Fall 2021

Design of Generic Enterprise Software:

Considerations for Cultivating a Knowledge Infrastructure

Jacob Kleivdal Ommundsen & Sigvart Bretteville-Jensen

2021

2021

Design of Generic Enterprise Software:
Considerations for Cultivating a Knowledge Infrastructure

http://www.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Generic enterprise software (ES) is often designed within software ecosystems, through a collaboration between a vendor, implementation partners and user organisations. Within software ecosystems, a key challenge for vendors is to develop knowledge about the practices and needs of a large set of user organisations to support the design of ES. The challenge arise as the user organisations are diverse, with heterogeneous interests and specific needs for the ES. Further, the diverse needs of user organisations are continuously evolving. Finally, the vendor does not have direct contact with many of the user organisations, as communication is often deferred to the implementation partners. Prior literature has described the set of arrangements employed by a vendor to develop knowledge about the practices and needs of their user organisations, as knowledge infrastructures. Yet, albeit its relevance for understanding design of ES, knowledge infrastructures remains an understudied phenomenon.

This thesis extends existing literature concerning design of ES by addressing the question: *How do vendors cultivate a knowledge infrastructure to support the design of generic enterprise software in software ecosystems?* Through a one year engaged research project, we have collaborated with the vendor of a global enterprise software, mainly used within health, and examined how they cultivate their knowledge infrastructure to support the design of ES. Our focus has been on examining the vendor's challenges in developing knowledge about their diverse set of user organisations, the collection of arrangements put in place to remedy these challenges and how this has evolved over time. Based on analysing our empirical findings, and a discussion with related academic literature, we identify four considerations for vendors when cultivating a knowledge infrastructure to support the design of ES. We see that vendors should consider: (1) Developing a generic mindset among implementation partners and user organisations, (2) Strengthening the absorptive capacity for externally produced knowledge, (3) Maintaining a distanced comprehension to user organisations and (4) Establishing strategic partnerships with user organisations. We argue that our empirical account and the four considerations identified, contribute to research and practice concerning design of generic enterprise software.

Keywords: Knowledge infrastructure, generic enterprise software, software ecosystems, cultivation

Acknowledgments

First and foremost, we would express our deepest gratitude to our supervisor, Magnus Li, for his guidance and support throughout our project. The countless hours of discussions and feedback have been invaluable, and this thesis would be nowhere near the final result without your help.

We would also like to thank the participants in our research project who took the time out of their busy schedule to talk to us about their work. Especially, we would like to thank the participants from HISP UiO and the members of the DHIS2 core team for your valuable input throughout our project. Additionally, we would like to thank our fellow students at the DHIS2 design lab for engaging discussions, laughs and two-hour lunches.

Finally, we would like to thank our friends and family for all their love and support. Jacob would especially like to thank his roommates at WT64 for the countless discussions around the breakfast table and great fun during his time as a master student. Sigvart would like to thank Elena for her support, patience and fantastic meals throughout the project.

Jacob and Sigvart
University of Oslo
December 2021

Table of contents

List of images

List of tables

List of figures

1 Introduction

Software implemented in organisations today are often generic enterprise software (ES), meant to fit and work across an array of organisational contexts. Common examples are Enterprise Resource Planning systems (ERP) and Electronic Health Records (EHR). As no two organisations are identical with varying practices, cultures and interests, a persistent challenge documented within the information systems (IS) research is designing ES that meets the diverse needs of user organisations (Soh, Kien, & Tay-Yap, 2000). Failure rates of ES implementations are high, with reports suggesting that 57% of implementations projects went over budget, while only 46% delivered on their expected benefits (Tan, Pan, Chen & Huang, 2020). Yet, the implementation of ES has become one of the most common organisational change events (Berente & Yoo, 2012) and yearly global spending on ES's was reported to be 2,5 trillion dollars in 2011, with an expectation of continued growth (Sykes & Venkatesh, 2017). Thus, with a growing market for ES and high failure rates of implementations, the design of ES represents an increasingly relevant context for IS research.

Literature has discussed several challenges related to the design of ES from the perspective of organisations implementing the solutions. A common theme is how generic software features of the ES clash with idiosyncratic needs of the individual user organisations (Strong & Volkoff, 2010; Berente et al., 2016; Davenport, 1998). As ES aims to fit across several organisational contexts, it is not possible for vendors to directly interact with user organisations and cater to their specific needs across all implementations (Titlestad et al., 2009). Rather, vendors often design ES as "half products", requiring configuration or customisation to the specific needs of the user organisation at the moment of implementing the solutions (Xu & Brinkkemper, 2007). To achieve this, design of ES is typically organised as a *software ecosystem* (Dittrich, 2014). The concept of software ecosystems describes the design, development, maintenance and implementation of an ES through collaboration between a software vendor, implementation partners and user organisations (Dittrich, 2014). Thus, the design of ES is often viewed as happening through two types of processes: The first is conducted by the vendor, designing the ES. The second is conducted by an

implementation partner, configuring and embedding the ES according to a user organisation's specific needs (Bansler & Havn, 1996, Schneider et al., 2018; Li & Nielsen, 2019).

Organising design of an ES as a software ecosystem can be beneficial in terms of ensuring a better fit between the ES and the idiosyncratic needs of each individual user organisation (Rickmann, Wenzel & Fischbach, 2014). However, it also requires significant resources from the vendor when supporting the design of the ES (Koch, 2007). Vendors must find means of developing knowledge about the practices and needs of the user organisations their ES is intended to serve. Yet, with an ecosystem of multiple implementation partners and a wide range of heterogeneous user organisations, developing knowledge to support the design of an ES is challenging for the vendor. In this thesis, *developing knowledge* refers to activities such as gathering, organising and processing information in a manner that is actionable towards supporting the design of an ES.

First, ES vendors need to consider not one, but many diverse user organisations with heterogeneous interest, all wanting their specific needs to be accommodated (Dittrich, 2014). As ES vendors cannot cater to the specific needs of each user organisation, they include features needed by many, while neglecting features only relevant to one or a few (Sia & Soh 2007; Kallinikos, 2004). Yet, literature also shows how a strategy of "design for everybody" might lead to "design for nobody", warning against a lack of focus on the diversity of needs (Oudshoorn, Rommes, & Stienstra, 2004). Secondly, as the diverse needs of user organisations are not fixed, but evolve, the vendors must strive to keep the ES useful and relevant to the user organisations over longer periods of time, rather than delivering it once (Dittrich, 2014). Therefore, developing knowledge to support the design of ES is an ongoing challenge for vendors. Third, the vendor does not have direct contact with many of the user organisations (Koch, 2007). Rather, this is often deferred to the implementation partners configuring and embedding the ES according to specific needs, as they operate in greater proximity to the user organisations (Dittrich, 2014). Consequently, the vendors also need to deal with implementation partners acting as intermediaries between them and the user organisations. Thus, organising an ES as a software ecosystem brings several challenges for the vendor in developing knowledge about their user organisations to support the design of the ES. Accordingly, a vendor needs to establish and sustain arrangements for

continually developing knowledge supporting the design of an ES. In this thesis, *arrangements* refer to social and technical practices, tools, and arenas that are used by a vendor to develop knowledge about the diverse set of user organisations to support ES design.

Early conceptualisations of how vendors develop knowledge to support the design of ES was described as "design from nowhere" (Suchman, 2002), either being based on assumed needs within a given market (Grudin, 1991), or arrangements focusing on indirect ways of knowing user organisations such as market-surveys (Bansler & Havn, 1996). Later research however, highlights how vendors have far more intricate arrangements supporting the design of an ES than prior literature would suggest (Pollock, Williams & D'Adderio, 2007). Pollock et al. (2007) highlight how vendors develop knowledge to support the design of an ES by actively searching for similarities among user organisations in order to lift specific needs out of their contexts and abstract them into "generic" software features. For instance, this is achieved through arrangements where user organisations are invited to negotiation meetings where the goal is aligning the needs between the participants (Pollock et al., 2007). Building on this work, Johnson et al., (2014) introduce the notion of knowledge infrastructures, which refers to the collection of arrangements that supports the vendor in the design of an ES. Jonhson et al. (2014) highlight how vendors have developed intricate knowledge infrastructures over time, describing a set of arrangements employed to develop knowledge about the diverse set of user organisations.

Both Johnson et al. (2014), Pollock et al., (2007) and Koch (2007) point toward that vendors invest significant efforts in cultivating a collection of arrangements (i.e a knowledge infrastructure) to support the design of an ES. Prior literature highlights various aspects related to developing knowledge about a diverse set of user organisations to support the design of ES. Yet, albeit its relevance and potential to further understand the process that goes into developing knowledge to support the design of an ES, knowledge infrastructures remains an understudied phenomenon. Based on this, we extend existing literature by exploring how a vendor cultivates a knowledge infrastructure.

1.1 Research question

We extend existing literature on design of ES within software ecosystems by addressing the following research question:

How do vendors cultivate a knowledge infrastructure to support the design of generic enterprise software in software ecosystems?

Knowledge infrastructures refer to the collection of arrangements used by a vendor to develop knowledge about the diverse set of user organisations to support the design of an ES. *Cultivation* refers to the process of vendors changing and adapting their knowledge infrastructure in an incremental and gradual manner to strengthen their ability to develop knowledge about a diverse and evolving set of user organisations.

In this thesis, we examine this question by first developing an understanding of the concept of knowledge infrastructure based on IS infrastructure theory. Secondly, with this understanding as a basis, we analyse the empirical findings from an engaged research project, where we have collaborated with the vendor of an ES named "DHIS2", focusing on how they have cultivated their knowledge infrastructure over time. Over a period of two decades, the DHIS2 software has grown from being an organisation-specific routine reporting system implemented in one country, to being an ES designed to support any case of gathering, management and analysis of health-related data. The DHIS2 software is now implemented in more than 70 countries and hundreds of organisations. In this regard, the vendor of DHIS2 can be considered as a highly successful case where a vendor has been able to cultivate a knowledge infrastructure that allows them to design software that sufficiently serves the needs of a vast set of user organizations. The engaged research project has been conducted through an interpretive case study, examining the vendor of DHIS2 challenges in developing knowledge, the collection of arrangements they introduced to address these (i.e their knowledge infrastructure), and how this has evolved over time. The aim of this project has been twofold. First, we aim to provide insight to the vendor of DHIS2, giving them an overview of their arrangements used to develop knowledge and how the relation

between the arrangements contribute to understanding their diverse set of user organisations.

Secondly, we have aimed to develop a contribution to academic literature relevant beyond DHIS2. Based on our empirical findings, our thesis offers two contributions to literature concerning design of ES within software ecosystems (Johnson et al., 2014; Silsand & Ellingsen, 2014; Dittrich, 2014; Gizaw et al., 2017; Joshi et al., 2007; Pollock et al., 2007). By extending the notion of knowledge infrastructure using concepts from IS infrastructure theory, we contribute with a conceptualisation for capturing a vendor's efforts in developing knowledge to support the design of ES. Further, by analysing how the vendor of DHIS2 has worked to develop knowledge supporting the design of the DHIS2 software over time, we identify four considerations for vendors in cultivating a knowledge infrastructure. We see that vendors should consider: (1) Developing a generic mindset among implementation partners and user organisations, (2) Strengthening the absorptive capacity for externally produced knowledge, (3) Maintaining a distanced comprehension to user organisations and (4) Establishing strategic partnerships with user organisations, when cultivating their knowledge infrastructure to support the design of ES. The four considerations extend the current understanding of the phenomenon of knowledge infrastructures by providing insight into considerations a vendor takes in cultivating their knowledge infrastructure, with the aim of supporting the design of ES. The four considerations may also act as a prelude to further research as each consideration presents interesting avenues that are not covered by our study. In addition to being relevant for researchers, our conceptualisations and empirical findings are relevant to practitioners grappling with developing knowledge about their diverse set of user organisations to support the design of their ES.

1.2 Chapter summary

Our thesis is further structured as followed:

Chapter 2: Related literature

This chapter describes literature related to design of ES within software ecosystems to provide an understanding of: (1) Design of ES in software ecosystems, (2) Three aspects for developing knowledge about a diverse set of user organisations to support the design of ES and (3) Knowledge infrastructures.

Chapter 3: Research approach

This chapter describes the background of our case, focusing on describing the DHIS2 ecosystem. It also elaborates on our engaged research project and chosen methodology - engaged scholarship and interpretive case study. Further, we describe our methods for data collection and the process of our analysis to highlight how we came to our contribution.

Chapter 4: Findings

This chapter describes our empirical findings. We present how the DHIS2 vendor's knowledge infrastructure has evolved from its starting point to its current state by looking at arrangements that have been employed to overcome the challenges of scaling the software globally.

Chapter 5: Analysis and discussion

This chapter starts by defining and discussing four considerations for cultivating a knowledge infrastructure to support the design of ES, and how they relate to findings from related literature. We further discuss the considerations relation to each other and how they serve different roles in enabling the knowledge infrastructure to support design of ES. Finally, we present how the four considerations contribute to both literature and practise on designing ES in software ecosystems.

Chapter 6: Conclusion

This final chapter summarises our thesis with a few concluding remarks.

2 Related Literature

The main goal of this thesis is to explore how vendors cultivate a knowledge infrastructure for supporting the design of an ES. The research builds on and aims to extend the stream of literature concerning design of ES within software ecosystems. In the first section, we will develop an understanding of the design of ES in software ecosystems. Second, we will highlight three aspects for developing knowledge about a diverse set of user organisations to support the design of ES. Lastly, we describe how we understand knowledge infrastructures in software ecosystems.

2.1 Design of Generic Enterprise Software

2.1.1 Generic enterprise software in software ecosystems

Generic enterprise software (ES) is defined as solutions designed for general use in a bigger market of user organisations, meant to fit and work across an array of organizational contexts (Bansler & Havn, 1996; Pollock & Williams, 2007; Silsand & Ellingsen, 2014). They are commonly understood as packaged software solutions used to support various business processes (Seddon et al., 2010). Common types of ES are enterprise resource planning (ERP) customer relation management (CRM) and decision support systems (DSS) which are used widely in large organisations (Magnusson & Nilsson, 2013). In contrast to bespoke, or "custom", software, ES represents a context for design that is characterized by significant variety and heterogeneity among user organisations (Sommerville et al., 2012). A persistent challenge documented within IS research is designing ES able to meet a diverse set of needs (Strong & Volkoff, 2010; Berente et al., 2016; Davenport, 1998). While it is desirable to cater to the specific needs of each user organisation, vendors cannot emphasise particularities across all of them. Thus, designing a one-size-fits-all cannot be achieved due to factors such as diverse practices, cultures and terminology existing within a large and heterogeneous audience of user organisations. (Soh et al., 2000).

One common strategy in combating this challenge is organising ES design as a software ecosystem. This strategy involves the vendor "opening" up its technology, allowing external organisations to configure, or extend, the ES according to user organisations specific needs

(Rickmann et al, 2014; Foerderer et al., 2019). As mentioned herein, software ecosystems describe the environment and relationship between the ES vendors and the affiliated organisations engaging in design, development and implementation surrounding a common software (Dittrich, 2014).

Hanssen (2012) highlights three main participants found within software ecosystems; i) A vendor acting as the keystone organisation, leading the design of the ES, ii) The user organisations of the ES and iii) The implementation partners configuring or extending the ES according to specific needs of user organisations. These networks of implementation partners and user organisations may cover a wide range of configurations, but a fundamental aspect is that their relations are based on a common interest in a central ES solution (Hanssen, 2012). Figure 2-1 illustrates how the design of an ES within a software ecosystem occurs, and the three main roles involved.

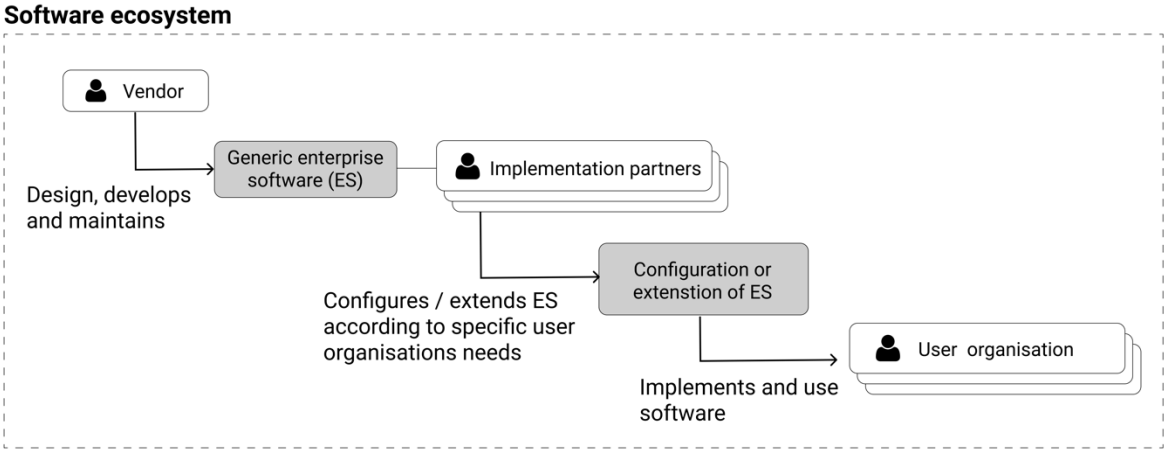


Figure 2-1: Design of ES within a software ecosystem

A precondition for vendors in successfully organising ES design as a software ecosystem lies within the possibilities for customisation and configuration in the ES. The ES must have the ability to be generic enough to fit a diverse set of needs, while being flexible enough to be configured and extended to the needs of specific user organisations (Light, 2005). In order for the vendor to strike this balance, Dittrich (2014, p. 1437) argues that: *"Continuous contact with users and customers is crucial. As the use contexts change, the evolution of software products to keep in sync with the application domain developments requires continuous effort"*. The vendor must, accordingly, continuously be able to take into account

the needs of a diverse set of user organisations when designing their ES (Dittrich, 2014). Thus, software ecosystems bring challenges for the vendor in developing knowledge about their diverse set of user organisations to support the design of the central ES. The next section will highlight three aspects related to developing knowledge about a diverse set of user organisations.

2.2 Aspect for developing knowledge to support design of an ES

The complex and evolutionary dynamics of a software ecosystem makes developing knowledge to support the design of ES an ongoing challenge for vendors. In this section, we describe three overall aspects for vendors when developing knowledge about their diverse set of user organisations to support the design of ES highlighted in extant literature: (1) Bridging a diverse set of needs, (2) Retaining the right distance (3) Communicating with the larger community of user organisations.

2.2.1 Bridging the diverse set of needs

The first aspect relates to the heterogeneity of user organisations existing within software ecosystems, and the challenge of bridging a diverse set of needs. Koch relates heterogeneity to the multi-spatiality of ES, writing: *"The systems and their vendors have developed into worldwide organisations, in which further development of software is occurring in literally hundreds of places in parallel.*" (Koch, 2007, p. 430). Around the common ES, one often finds a vast group of heterogeneous user organisations, all having their own specific needs (Dittrich, 2014). As the vendor cannot accommodate all specific needs, they are often relying on arrangements focusing on the alignment of diverse needs in order to develop knowledge about the diverse user organisations to support the design of an ES (Ulriksen, 2017). This implies shifting the focus of design to the general needs, rather than a focus on the particularities of each user organisation (Sia & Soh, 2007; Strong & Volkoff, 2010).

Early research on design of ES, conceptualised the process of developing knowledge to support the design of an ES as "design from nowhere". This is based on a nearly technocratic ideal with little, if any, contact with the user organisations of the ES (Suchman, 2002). The need for user involvement and developing knowledge about local work practices were highlighted as important factors for the design of ES (Bansler & Havn, 1996). However, it

was argued that to “mass produce”, vendors mainly targeted the “greatest common denominator” in a given market, relying on indirect ways of knowing the user organisations, such as market surveys, with the end goal of making the ES as widely marketable as possible (Bansler & Havn, 1996; Grudin, 1991).

Later research has, in contrast to earlier perspectives, shown that the vendor’s arrangements for developing knowledge is far more intricate than first noted. Coining the term “design fallacy”, Stewart & Williams (2005) captured the presumptions made in several fields of literature, in which critical accounts of “technocratic” design activities, based on inadequate and misleading views of user organisations needs were highly present. Taking the argument further, Koch (2007) and Pollock et al., (2007) noted how these accounts lagged far behind industry practices. They argued that ES are not “design from nowhere”, as some earlier research suggested (Suchman, 2002; Bansler & Havn, 1996). Rather, the design of ES was based on aligning diverse needs to develop the necessary knowledge to support the design of ES. Extending the debate on how technology was built to work across a diverse range of settings, Pollock et al. (2007) introduced the concept of generification.

The concept of generification describes a set of mechanisms used to develop knowledge for the design of an ES, where specific needs are lifted out of their context and abstracted into ‘generic’ features through alignment work (Pollock et al., 2007). They highlighted how this was done by moving the process of deciding on software features for an ES to public forums, focusing on a process of negotiation where the goal is aligning diverse needs between a wide set of user organisations (Pollock et al., 2007). Generification strategies have further gained some traction in extant literature, describing how vendors are able to bridge the gap between a heterogeneous set of user organisations. Johannesen & Ellingsen (2009) describes how generification strategies were used to support design when taking an existing bespoke system and making it into an ES. Other studies have also discussed and exemplified how generification processes have supported the design of health information software (Gizaw, Bygstad, & Nielsen, 2017). Some variations on the concept of generification exist. However, the commonality is describing how vendors focus on negotiation and dialogue between user organisations, intending to align their diverse needs into a common set of generic features. As such, vendors can effectively develop knowledge to support the design

of ES. Accordingly, the curbing or "alignment" of the diverse set of user organisations needs is an important part of the vendor's work in developing knowledge to support the design of ES (Pollock et al., 2007).

2.2.2 Retaining appropriate distance

The second aspect unpacked in extant literature is the distance existing between the vendor and the user organisations. Distance refers to the social relation the vendor keeps to the user organisations, where a low distance refers to a close connection while a high distance refers to the opposite. As developing knowledge about the user organisations within an ecosystem requires some efforts towards alignment work, specific needs of each setting have been labelled as barriers to development (Koch, 2007; Pollock et al., 2007). As such, vendors of ES intentionally keep a certain distance, fearing that too close affiliations with a single user organisation will lead to their solutions becoming too specialized, and therefore not usable or relevant to other organisations (Bansler & Havn, 1996; Koch, 2007, Sawyer, 2001).

Distance can also emerge between vendors and user organisations unintentionally due to the growing number of user organisations in the software ecosystem as a whole (Johnson et al., 2014). This constitutes a somewhat diverging need for the vendor in the process of developing knowledge to support the design of ES. At the one end, it puts requirements on the vendor to continuously be able to take new organisations and contexts into account, accommodating the increasingly distributed settings (Roland et al., 2017). A task that might require more direct engagement with new user organisations (Johnson et al., 2014, p. 806). At the same time, an intentional distance might be required to develop knowledge "generic" enough for the solutions to remain relevant across many contexts. The challenge is to balance between necessary forms of direct engagement and maintaining a distance. In discussing how vendors make sense of their diverse set of user organisations, Campagnolo et al., highlights this tension, writing: *"In order to produce knowledge that extends in space-time, knowing how it may become possible not to have to interact, we argue, is as relevant as knowing how to interact."*. (2015, p. 158). Campagnolo et al. (2015) highlight that more attention should be given to how vendors retain appropriate distance to the user organisations. They argue that this is achieved through alignment work such as (1) encouraging participants to carry out organisational change to align with the system, (2)

turning them into 'sellers' who can encourage other user organisations elsewhere to align their specific needs with other user organisations and (3) identify selected user organisations as benchmarks for creating "best practice software".

It is apparent that various efforts of alignment work are ways for vendors to maintain a level of distance to the user organisations. However, some challenges might be pointed out with this strategy. A commonality between the alignment efforts conceptualised is that a small group of intermediaries, representing different user organisations, are central in communicating the needs of the larger "community" of user organisations (Fruijtier & Pinard, 2017). As highlighted in Johnson et al. (2014) and Fruijtier & Pinard (2017), these intermediaries are able to increase their influence in these processes over time. Thus, alignment of diverse needs through the use of intermediaries is prone to power discrepancies. As the position of the intermediaries in the alignment processes can vary (Pollock et al., 2007; Johnson et al., 2014), the critique is then that one could end up with a situation where the base of knowledge vendors use to design the ES, is not truly reflective of the diverse needs within the ecosystem (Fruijtier & Pinard, 2017). One such example is described by Wagner et al., (2006), highlighting a case whereby the vendor chose a small group of user organisations as the only source of information to develop knowledge for designing an ES. This little group then came to define "the best practice" for an entire industry, leading to several challenges upon implementing the ES in other user organisations. Other literature has also exemplified how a lack of focus on the diversity of needs affords challenges in the process of design of ES (Oudshoorn et al., 2004).

In summary, extant literature has argued that vendors need a more persistent focus on the local work practices to ensure that the solutions do not directly contradict the existing practices of the user organisations (Joshi et al., 2007) However, an intentional distance might also be required to support the design of an ES with a base of knowledge that applies to the scope of contexts. Retaining appropriate distance is by many argued to occur through forms of alignment work where the ES vendor develops knowledge by filtering out, curbing and aligning needs to develop knowledge that supports the design of an ES (Pollock et al. 2007; Johnson et al. 2014). While alignment work is central, contextual knowledge has also proven relevant for vendors in this context. Thus, to develop knowledge that best supports

the vendor in the design of ES, a vendor needs to balance between necessary forms of engagements and retaining an appropriate distance.

2.2.3 Communicating with the larger community of user organisations and implementation partners

The third aspect mentioned in extant literature regarding developing knowledge to support the design of ES, is communicating with the larger community of user organisations. A commonality in the articles highlighting arrangements used to develop knowledge to support the design of ES is that the main points of interaction are happening between the vendor and intermediaries, often being representatives from implementation partners, communicating the needs on behalf of the community of user organisations (Pollock & Williams, 2007; Silsand & Ellingsen, 2014; Johnson et al., 2014; Fruijtier & Pinard, 2017). However, literature has also highlighted other forms of contact to be important.

In response to alignment efforts and intentional distance kept by vendors, implementation partners and user organisations tend to create their own solutions in response to dissatisfaction with their needs not being met (Von Hippel, 2005). As such, prior literature has highlighted these innovation activities within the community of implementation partners and user organisations as important sources of knowledge relevant to the vendor's design processes (Fruijtier & Pinard, 2017; Gizaw et al., 2017). This is especially relevant in the context of software ecosystems, highlighted by Dittrich writing: *"Innovation takes place across the whole ecosystem. Contact with users and other actors is therefore important to keep the innovative edge of the software product"* (2014, p. 1455). As parts of the design of an ES within ecosystems is often deferred to implementation partners closer to the concrete contexts of use, these represent an important source of potentially valuable knowledge. Gizaw et al., (2017) concur, arguing that processes of extracting knowledge from local contexts to the more global contexts should not only be based on selected requirements by the vendor, but also the innovations outputs created within these local contexts.

While innovation within the larger community of implementation partners and user organisations can be an important source of knowledge, Mozaffar (2016) highlights this as a default assumption made by extant research, potentially neglecting other roles these could have within the ecosystem. The paper highlights the importance of going beyond the

singular view of “the user”, acknowledging that association and linkages exist between diverse participants within a software ecosystem. The paper argues that the larger community within the ecosystems should rather be seen as a space of information and knowledge exchange between various groups, covering a wide range of activities beyond user innovation (Mozaffar, 2016, p. 237). Central to this space of information and knowledge exchange is the coordination of the heterogeneous interests of different user organisations and implementation partners from different constituencies, where activities such as developing a common voice and being an up-to-date informant are present (Mozaffar, 2016). Thus, communicating with the larger community of implementation partners and user organisations is an important aspect of the vendor's work in developing knowledge to support the design of ES.

2.3 Understanding knowledge infrastructures in software ecosystems

Till now, we have described how the design of an ES within a software ecosystem occurs, the participants involved in the processes, and highlighted three aspects for vendors when developing knowledge about their diverse set of user organisations. In this section, we introduce a set of concepts, highlighting how we understand the process of developing knowledge about a diverse set of user organisations to support the design of an ES. To explore the collection of arrangements used by a vendor to develop knowledge about the diverse set of user organisations, we utilise the concept of knowledge infrastructures (Johnson, 2014). Further, we use concepts from information infrastructure theory (Hanseth & Lyytinen, 2010) and see the development of knowledge infrastructures as an evolving process happening through cultivation.

2.3.1 Arrangements, knowledge infrastructures and cultivation

As mentioned herein, arrangements refer to social and technical practices, tools, and arenas that are used by a vendor to develop knowledge about the diverse set of user organisations. Examples of arrangements mentioned in prior literature include specific methods such as market surveys, meetings with user organisation representatives and usability feedback workshops (Johnson et al., 2014). We refer to the collection of arrangements as knowledge infrastructures, supporting the vendor in the design of an ES. (Johnson et al., 2014).

The design of an ES seeks to support both implementation partners, configuring and customising the ES, and the diverse set of user organisations (Dittrich, 2014). These contexts are not static and as such, vendors need to juggle the diverse needs emerging from these heterogeneous user organisations, striving to accommodate the different contexts and areas of use. As these contexts change, the vendor strives to change and adapt the knowledge infrastructure supporting the design correspondingly to sustain the ES usefulness and relevance. Thus, the ES vendors continuously work towards developing their collection of arrangements - the knowledge infrastructure, to support the design of an ES.

Figure 2-2 illustrates our conceptual understanding of developing a knowledge infrastructure. As shown, we see this as an evolving process happening through cultivation (Grisot et al., 2014). Software ecosystem contains a heterogeneous set of user organisations with diverse needs and the many customisations made by implementation partners, which are everchanging (Monteiro et al., 2013). Thus, similar to information infrastructures, developing a knowledge infrastructure is not a static one-sided endeavour constructed by the vendor nor is it designed by selecting and putting together arrangements based on fixed goals with planned outputs (Bergqvist & Dahlberg, 1999; Hanseth & Lyytinen, 2010). ES vendors must rather attempt to change and adapt the combination of their arrangements (i.e the knowledge infrastructure) in an incremental and gradual manner according to the changes happening within the ecosystem. Viewing the development of a knowledge infrastructure as cultivation draws attention to how the process is shaped and influenced by the dynamics within the software ecosystem, either enabling or constraining the possibilities of change (Grisot et al., 2014). As such, a vendor can only continuously cultivate the knowledge infrastructure to improve their ability to develop knowledge about the diverse set of user organisations.

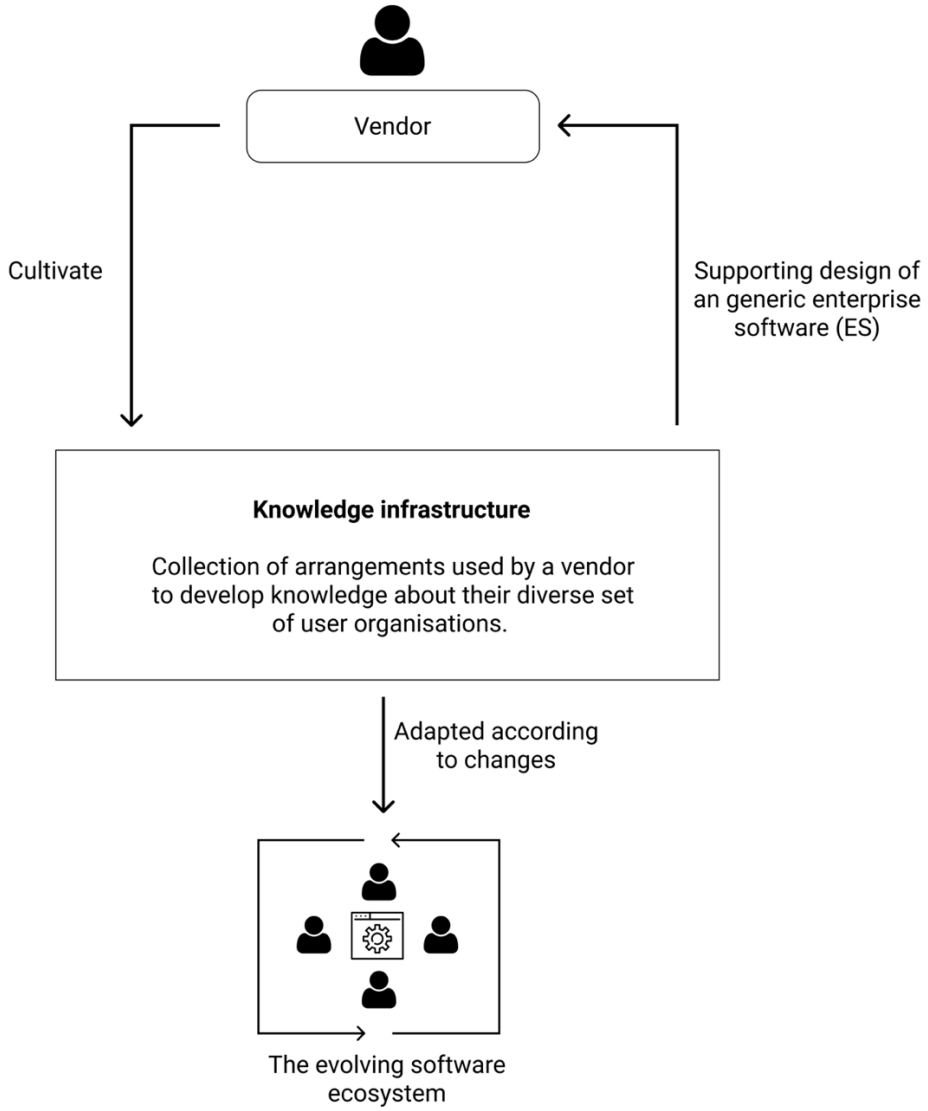


Figure 2-2: How a vendor cultivates a knowledge infrastructure

2.4 Chapter summary

To support the design of an ES, vendors must develop knowledge of their diverse set of user organisations. We refer to the collective of arrangements used by a vendor to develop knowledge about the diverse set of user organisations as knowledge infrastructures. Knowledge infrastructures are subject to ongoing cultivation by the vendor to improve their ability to develop knowledge about the diverse set of user organisations. Cultivation refers

to the process of vendors changing and adapting their knowledge infrastructure in an incremental and gradual manner according to the changes happening within the ecosystem. Table 2-1 summarises the three aspects for developing knowledge about a diverse set of user organisations to support the design of ES, highlighted by extant literature. Albeit its relevance to further understand how vendors manage these aspects when developing knowledge to support design of an ES, knowledge infrastructures remains an understudied phenomenon. Based on this, we extend existing literature by exploring how a vendor cultivates a knowledge infrastructure.

Aspect	Description
Bridging a diverse set of needs	Heterogeneity of user organisations affords challenges in developing knowledge. Alignment work, focusing on negotiation and alignment of diverse needs, has gained traction in describing how vendors develop knowledge to support the design of an ES (Pollock et al., 2007).
Retaining appropriate distance	Alignment work labels specific needs as barriers to development. Thus, vendors keep a distance, fearing too close of affiliations with single user organisations (Sawyer, 2001; Koch, 2007; Campagnolo et al., 2015). Yet, a stronger focus on the diversity of needs as well as the local work practices of user organisations has also been advocated for (Joshi et al., 2007; Oudshoorn et al., 2004).
Communicating with the larger community of user organisations and implementation partners	Communication is mainly happening through implementation partners (Silsand & Ellingsen, 2014; Johnson et al., 2014). Yet, activities happening within the larger community of user organisations and implementations partners is highlighted as sources of potentially valuable knowledge, both relating to innovation outputs (Fruijtier & Pinard, 2017; Gizaw et al., 2017) and for keeping updated on the needs and challenges of diverse user organisations (Mozaffar, 2016).

Table 2-1: Three aspects for developing knowledge about a diverse set of user organisations

3 Research approach

In this chapter we outline our research approach. The empirical research of this thesis is based on a one-year long research project in collaboration with the vendor of the District Health Information Software 2, referred to as "DHIS2". The chapter is organised in the following manner: First, we provide our case description of the DHIS2 and the categories of participants that form the ecosystem around the DHIS2 software. Second, we describe our research methodology and how we have conducted our research project in collaboration with the vendor of DHIS2. Third, we present the various forms of data gathering used to gather our empirical findings. Lastly, we describe the process of our analysis to highlight how we came to our contribution.

3.1 Case description: HISP and DHIS2 software

This study is part of an ongoing global research program, named the Health Information Systems Programme (HISP), centred at the University of Oslo. Since 1994, the program has been involved in strengthening health information systems in low- and middle-income countries. The overall goal of HISP is to *"enable and support countries to strengthen their health systems and their capacity to govern their Health Information Systems in a sustainable way to improve the management and delivery of health services"* (UiO, n.d.-a). At the center of the HISP project is the development of the generic enterprise software in focus of this paper, DHIS2. DHIS2 is an open-source, web based solution, mostly used as an ES for management of health related data (DHIS2, n.d.). DHIS2 is designed and developed to support the gathering, management, storage and analysis of health related data. In its origin, the DHIS2 software started as a standalone application that was built to address the challenge of fragmented health management software solutions occurring in post-apartheid South Africa (Braa & Sahay, 2017). However, the initiative has evolved into a global operation with both research and implementation happening globally. At the time of writing DHIS2 is implemented in more than 70 countries around the world, making it the world's most widely used ES for health management (DHIS2, n.d). Image 3-1 illustrates the global footprint of DHIS2.

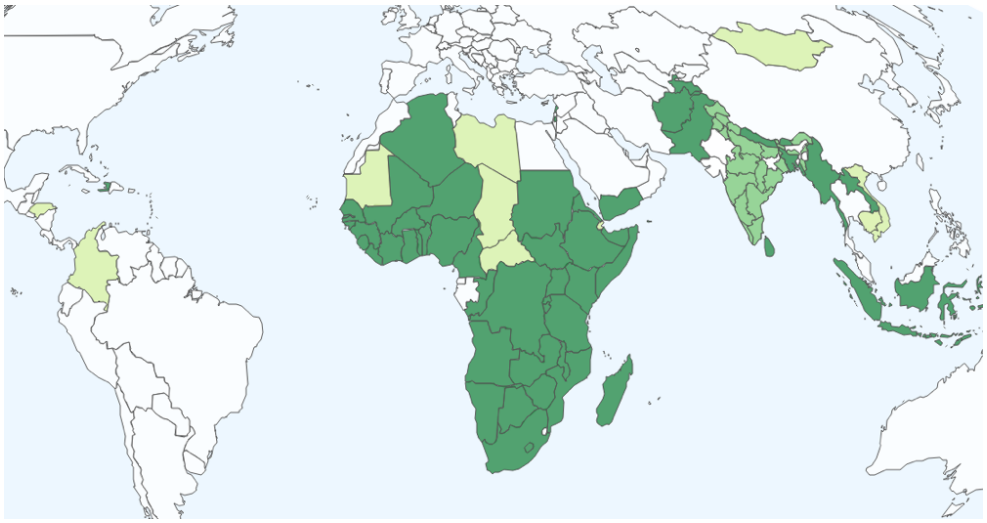


Image 3-1: DHIS2 global footprint

In order to manage this rapid growth, software development of DHIS2 has evolved into being structured similar to Dittrich's (2014) description of software ecosystems. In the following section we will highlight the different categories of participants, and their roles, that form the ecosystem around the DHIS2 software. This is further referred to as the *DHIS2 ecosystem*.

3.1.1 DHIS2 ecosystem and its participants

The DHIS2 ecosystem draws similarities to Hanssen's (2012) description of the three main roles participating in the software ecosystem; (1) A vendor, (2) Implementation partners and (3) User organisations.

The first role is the DHIS2 core team, acting as the vendor of the DHIS2 software. The core team is located at the University of Oslo and is mainly in charge of design and development of the generic DHIS2 software. The core team is made up of 80 staff members, a relatively modest number compared to the 2.4 billion people who live in the countries where DHIS2 is used. It consists of researchers, developers and managers that together coordinate the design and development of the DHIS2 software.

The second role in the DHIS2 ecosystem are the HISP groups, acting as the implementation partners. The HISP groups are a global network of regional partners located in either one or multiple countries around the world, examples being India, Malawi, Mozambique, Nigeria and Rwanda, totalling a number of 16. The HISP groups are consultancy firms, specialising in implementing the DHIS2 software according to the specific needs of user organisations, either through configuration or extension of the generic DHIS2 software. In addition, the HISP groups aid the DHIS2 core team in gathering needs from user organisations, capacity building within the region, and generally strengthening DHIS2 as an ES.

The third role are the ministries of health and non-governmental organisations (NGO) which can be categorised as the user organisations of the DHIS2 software. The user organisations cover a wide range of different types of organisations, using the DHIS2 software within a widely different scope. DHIS2 implementations may range from NGO´s such as the World Health Organisation (WHO) operating at an international scale, to ministries of health in Rwanda. Figure 3-1 illustrates the participants within the DHIS2 ecosystem and their relation.

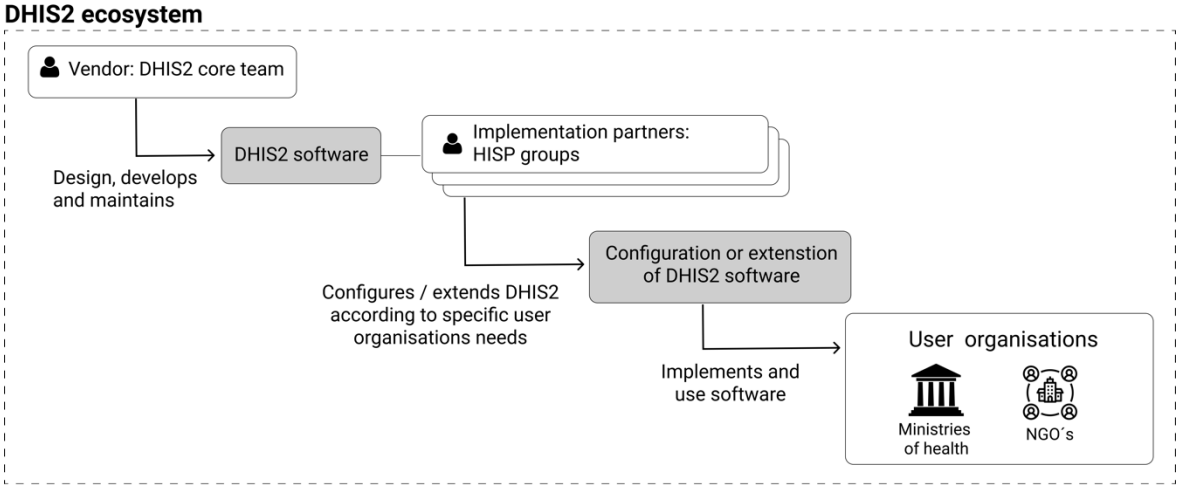


Figure 3-1: The DHIS2 Ecosystem

As figure 3-1 shows, the vendor (DHIS2 core team) designs and develops DHIS2 as a generic ES. This is further configured and extended by implementation partners (HISP groups) to fit the various user organisations (e.g. Ministries of health, NGOs). In the next section we will elaborate on how we have conducted our research.

3.2 Research methodology

Before we turn to the details of our data collection and analysis, we will describe how we have conducted our research project and the main activities involved. First, we describe our introduction to the research project and how we came to conduct an engaged scholarship. Following, we describe our research process centered around three main activities; 1) Researcher-practitioner negotiation, 2) Problem-formulation and 3) Selecting a form of inquiry.

3.2.1 Introduction to the project and engaged scholarship

Our introduction to the project came through our connection with the DHIS2 design lab. The design lab is a part of HISP UiO and is a research group consisting of post-graduate students from the department of informatics at the University of Oslo. The group is focused on exploring ways to support the design and innovation of DHIS2 (Li, 2019), often through forms of engaged research (Mathiassen, 2017). The design lab has since 2018 been closely connected to the DHIS2 core team through previous and ongoing collaborative research projects. This close connection has facilitated a context where research and practice work together in strengthening the design and innovative capacity of DHIS2 (UiO, n.d.-b). As members of the design lab, we were provided with access to collect empirical data, but also with an opportunity to collaborate with the core team in a joint research-practitioner project. This led to a dialogue with members of the core team, where discussions of engaged scholarship (Mathiassen, 2017) emerged. In engaged scholarship, researchers work with practitioners to address a real-world problem situation while contributing to academic literature (Mathiassen & Nielsen, 2008). It is characterised as a participative form of research focused on obtaining different perspectives from key stakeholders (researchers, clients and practitioners) in studying complex problems (Van de Ven, 2007). Conducting an engaged research project has been the overall strategy for both identifying and answering the research question presented in this thesis. Following, we will describe our engaged

research project centred around three main activities: 1) Researcher-practitioner negotiation, 2) problem-formulation and 3) selecting a form of inquiry. The engaged research project has been conducted in an iterative manner, whereby these three activities have been revisited several times as the project evolved. The form of inquiry has been selected, and continuously assessed, based on both the relevance to address the evolving problem formulation and feasibility in the collaboration with the DHIS2 core team. Image 3-2 from Li (2021) illustrates the three main activities of an engaged research project and their relation.

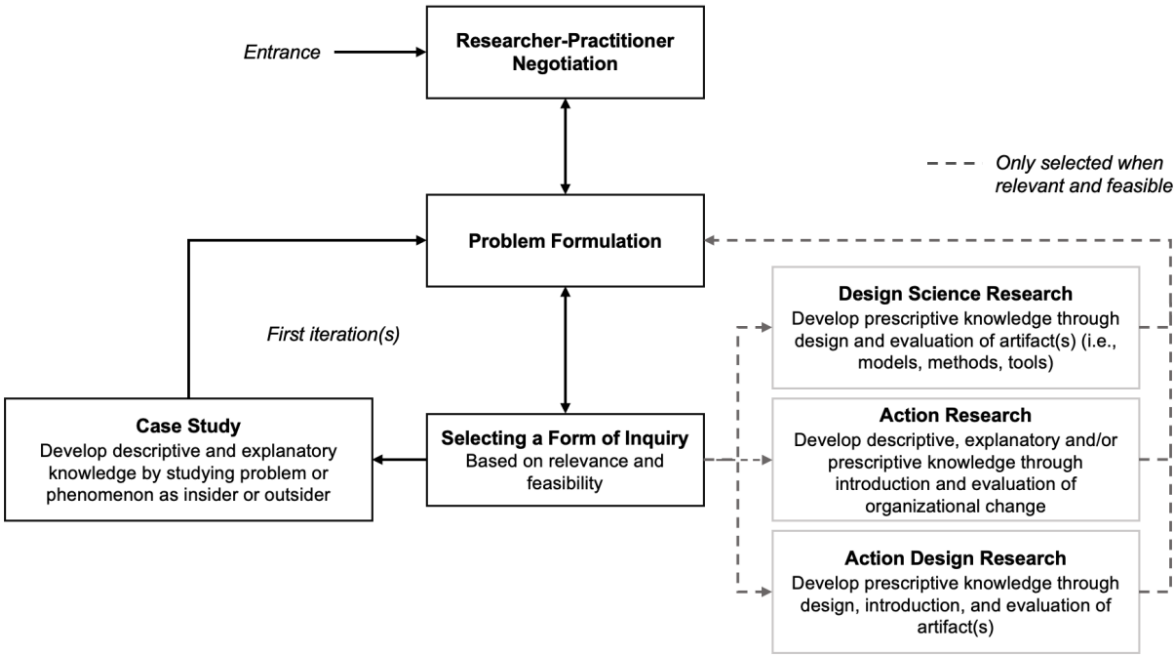


Image 3-2: Main activities in engaged research from Li (2021)

3.2.2 Researcher-practitioner negotiation

Our entry point to the engaged research project was an initial research-practitioner negotiation, where we engaged in a dialogue with members of the core team to discuss the scope of the project and lay the foundation for further inquiry. We entered the collaboration with an interest and curiosity of potentially understanding more of the DHIS2 core team's inner workings and how they managed to make an ES that was used on a global basis. Through the initial dialogue, we identified the mutual research interest of focusing on the

DHIS2 core team's practices for designing the DHIS2 software. The initial problem area of interest was vaguely formulated, but we agreed to mainly focus on the DHIS2 core team's role as a vendor and their practices for designing DHIS2. This initial problem area was later reformulated based on several iterations of data gathering. We highlight how this has evolved in the next section.

The initial research-practitioner negotiation also included defining the practicalities of our collaboration. We agreed on access to potential informants for interviews and other forms of data gathering. We also defined our roles in relation to the DHIS2 core team, which have remained unchanged throughout the project. While we have aimed to be engaged in a context, collectively defining the research problem and understanding it through the perspective of the informants, we as researchers were not immersed in the context through active participation in the DHIS2 core teams daily activities. Our participation throughout the project can rather be seen as "detached outsiders" (Van de Ven, 2007) where we have acted as impartial onlookers, gathering empirical material from a broader set of informants, continuously aiming to address the evolving problem formulation. We have had control of all research activities, with the informants taking an advisory role in terms of formulating the problem and providing feedback (Van de Ven, 2007).

This being said; It is important to note that "detached" does not necessarily equal being unbiased or fully objective. Rather, we argue that our positionality of "detached outsiders" can be compared to Walsham's (2006) description of "neutral". Walsham (2006) views positionality as a spectrum, ranging from the neutral observer to the full action researcher, potentially changing over time.

3.2.3 Problem-formulation

The second activity that has been continuously revisited throughout our engaged research project has been the framing and formulation of our research problem. The framing and formulation of our research problem has been gradually modified, elaborated and specified through an iterative process, informed both by findings from empirical inquiries and related concerns in academic literature. It has been a back and forth process of capturing the dual knowledge interest, engaged research projects seeks to address. Both producing knowledge

relevant to the real-world problem situation of the DHIS2 core team, as well as producing knowledge relevant to a related concern in academic literature (Mathiassen, 2017).

The initial problem area identified the mutual research interest of focusing on the DHIS2 core team's role as a vendor and their practices for designing the DHIS2 software. Based on several rounds of data gathering, focusing on understanding more of the DHIS2 core team challenges in designing the DHIS2 software, the problem formulation was revisited. In collaboration with members of the DHIS2 core team, we identified a more specific interest for an overview of the various arrangements for developing knowledge to support the design of DHIS2 present in the organisation. Their practices were continuously evolving, and the core team members explained that no single person had an overview of how the process occurred. They were also interested in feedback for potential improvement of the current ways in they were developing knowledge about their diverse set of user organisations. In parallel of gradually framing and understanding the real-world problem situation of the DHIS2 core team, we identified a similar concern in the extant literature related to design of ES. Specifically, we found that research had argued that industry practices for developing knowledge about a diverse set of user organisations are far more intricate than extant literature would suggest. This initial link between the real-world problem of the DHIS2 core team and the concern in extant literature informed further data collection and analysis, as well as guided the iterative problem formulation phase throughout our project.

The final outcome of this iterative problem formulation, informed both by findings from empirical inquiries and related concerns in academic literature was the research question addressed in this thesis: *How do vendors cultivate a knowledge infrastructure to support the design of generic enterprise software in software ecosystems?* One the one hand, the question is grounded in the "real-life" situation of the DHIS2 core team, addressing their challenging effort to develop knowledge for supporting the design of DHIS2. On the other hand, it relates to the concern expressed in academic literature, attempting to produce knowledge relevant for literature concerning design of ES within software ecosystems. How the problem formulation evolved is described in greater detail in section 3.4 where we describe our process of data analysis.

3.2.4 Selecting a form of inquiry: Case study

The third activity that has been continuously revisited throughout our engaged research project has been to assess the most relevant and feasible form of inquiry, referring to the methodology, guiding our research project. As the core commitment of an engaged research project is to contribute to both practice and theory, it can draw on a variety of forms of inquiry to guide the research (Mathiassen, 2017). The form of inquiry has been selected, and continuously assessed, based on two concerns; 1) relevance to address the identified problem and research question and 2) feasibility given our collaboration with the DHIS2 core team (Li, 2021).

During the early stages of the research project, the form of inquiry was similar to that of a case study. Here, we focused on gaining a general understanding of the DHIS2 core team and their inner workings as "detached outsiders", gathering data from a variety of different members within the core team (Van de Ven, 2007). As the project evolved, we continued to assess case study as a relevant form of inquiry, and considered our options to change towards more intervening forms of inquiries such as action research (Mathiassen, 2017). Based on the two concerns of relevance and feasibility, case study remained our form of inquiry throughout the research project.

Over the course of our research project, the problem formulation evolved towards a focus on how the DHIS2 core team was developing knowledge about their diverse set of user organisations to support the design of DHIS2. A viable path to answering this was further understanding the challenges in, and arrangements used to, develop knowledge in order to support the design of DHIS2. This focus resonated with the case studies, described as a detailed inquiry of a specific case with a focus on the activities, functions and local meanings within this case (Stake, 2005). As the process of developing knowledge about their diverse set of user organisations involved a variety of different members within the core team, the need for gathering empirical data on a broad set of aspects and perspectives remained relevant to our defined problem. Additionally, as the problem formulation further evolved to include how their set of arrangements for developing knowledge had evolved over time, the need for a broad perspective persisted. Thus, conducting a case study as "detached outsiders" (Van de Ven, 2007) remained relevant, gathering our empirical knowledge

through interviews, observations and document analysis to ensure a broader perspective. This has allowed for gaining an holistic understanding of the identified problem, rather than becoming too embedded in a specific area. Secondly, while we throughout our case study identified some challenges that potentially could be relevant for a form of inquiry based on more systematic intervention such as action research (Mathiassen, 2017), case study persisted as most feasible due to the timeframe and scope of our project.

The philosophical foundations of our research can be classified as interpretive (Klein & Meyers, 1999). Our epistemological stance has not been based on trying to explore the research question through quantifiable metrics. The focus has rather been to investigate the social phenomenon of developing knowledge to support the design of DHIS2, attempting to understand it through the intersubjective meanings and experiences of our informants (Myers, 1997). Bringing meaning to such organisational settings, where social and technological complexity exists in an evolving space, is thus not a task in search of some objective truth. This context is a social construction, influenced and impacted by the human sense making that goes into this process and thus not necessarily objectively "observable". As argued by Crang & Cook (2007), it is the way in which people make sense of the events around them that tells us something about how a view of the world is constructed, understood and acted upon. As such, understanding this phenomenon without taking into account the intersubjectivity that gives it meaning would arguably make it difficult to produce a valuable contribution.

3.3 Data collection

In this section we present the various forms of data gathering activities conducted during this research project. We will mainly highlight three data gathering activities and argue for their relevance in regards to our project.

3.3.1 Interviews with the DHIS2 core team

Our primary source of information through the project has been interviews and meetings with members of the DHIS2 core team. In total we had 16 interviews with various core team members. Our choice of interviewees were based on attaining a variance of perspectives to best understand the real-world problem situation. Accordingly, we had interviews with four

different roles ranging from senior researchers, coordinating the long term strategy of the DHIS2 software, to core team developers concerned with feature development. As our engaged scholarship needed continuous contact with the context we were engaged in, we established a few contact points within the DHIS2 organisation that we met with throughout the whole research project. This allowed us to gradually grasp and build up an understanding of the complex context and the real-world problem we were engaged in. These meetings were accompanied with longer interviews where we went in depth on specific topics with them as well as other members of the core team.

Ten of the interviews we conducted were with "product managers", who are in charge of one part of functionality within DHIS2. Their responsibility relates to managing the day-to-day design process of their given part. They are also responsible for developing knowledge to support the process of designing this part, through arrangements for understanding the user organisations. As we were interested in investigating the arrangements the DHIS2 core team used for developing knowledge, interviewing the managers responsible for these arrangements was also one of our main sources of information. These product managers also became our contact points within the organisation and provided us with a basis for the context we were investigating, but later also provided new information during our continuous meetings throughout the project.

Additionally, we had three interviews with DHIS2 researchers who also have managerial responsibilities, but majorly partake in deciding the long term strategy of DHIS2. These people have been with DHIS2 for a long time and provided good input on how the software has changed in the last 20 years. This was especially useful in the later stages of the project when our focus was on understanding how the knowledge infrastructure has evolved from a one-country implementation to becoming a global ES.

Four core team developers were also interviewed as they provided another perspective on the real-world problem situation that we were engaged in. We wanted to understand the challenges and views of the people working directly with developing DHIS2, as we were not only interested in the top-down perspectives of managers and coordinators. Among the core team developers interviewed was one technical leader, responsible for a team of developers.

These interviews were specifically relevant in the early exploratory phase as they provided concrete challenges related to how knowledge was put to use when designing DHIS2.

3.3.2 Observations of meetings and digital platforms

To enrich our understanding we combined interviews with being a neutral observer in relevant meetings within the DHIS2 core team. Specifically, we partook in meetings regarding evaluations of the current arrangements for developing knowledge about user needs. These were valuable sources of information as they gave insight to multiple views and experiences both the core team members, as well as the HISP groups, had with the current arrangements. The meetings were conducted in a manner where different people would share their personal views on how they thought a specific arrangement was working, others then commented, and discussions occurred. These were also digital meetings where we as observers became almost invisible and it seemed like the meeting participants became unaware of our attendance and thus the data gathered was more raw and uncurated than in our interviews.

In addition, we were also allowed in as observers on the DHIS2 internal Slack-channels. Slack is a collaboration software where you can indulge in conversations and discussion with anyone in the company at any given time, similar to a giant chat-room. In companies like DHIS2 with user organisations and HISP groups spread globally, Slack is used to keep anyone up to date and have a common place to share information and updates. Also here we became observers as Slack gave rich insight on the everyday ongoing of the DHIS2 core team. These observations supplemented the interviews with multiple perceptions that can be used to clarify meanings by identifying different ways the case is being seen, verifying the repeatability of our interpretations (Stake, 2005, p. 454).

3.3.3 Document analysis

A third and final source for data gathering was document analysis. As part of our introduction to DHIS2, we were given documents that were used to externally describe the organisation to people unfamiliar with the project. Additionally, much information exists on websites like DHIS2.org regarding partners, news and research. DHIS2 also has online forums such as the community of practice where implementation partners, user organisations and core team members discuss issues and provide each other with support.

All of these were sources to our document analysis which was an important tool to become familiar with the case we were engaged in.

Table 3-1 summarises the data gathering activities conducted during this one year project. As shown, we have had interviews with a variance of core team members as well as observed meetings the core team have had throughout the project. The table is missing our continuous efforts to also follow the community of practice and our participation in the internal Slack-channels. It is the combination of these methods that makes up our empirical grounding which we further use for the data analysis. This will be described in the next section.

#	Role of participants	Type of data gathering	Participants	Quantity	Date
1	DHIS2 product managers	Interview	2	10	26.09.20 - 05.03.21
2	DHIS2 core team developers	Interview	4	4	10.03.21 - 15.03.21
3	DHIS2 researchers	Interview	3	2	26.03.21 - 05.05.21
4	DHIS2 core team meetings	Observation	14	2	10.10.2021
Total			23	18	26.09.20 - 10.10.21

Table 3-1: Overview of methods for data gathering

3.4 Data Analysis

The analysis of this thesis is a parallel process of engaging in both the literature and empirical data, allowing for an abductive process where our contribution was iteratively shaped by both. Based on a thematic analysis of our empirical data (Braun & Clarke, 2006) and by developing a theoretical framework from relevant literature streams, we have combined inductive and deductive reasoning to contribute both to practice and literature. Over the course of a year, our research project has been through a change process which we divide into two parts: i) Understanding the DHIS2 ecosystem ii) Forming our contribution.

3.4.1 Understanding the DHIS2 ecosystem

3.4.1.1 Initial coding, categorisation and thematic analysis

The problem formulation phase in our research project pointed to a need within the DHIS2 core team to better the overview of how the core team develops knowledge to support the design of DHIS2. In order to gain an overview, we wanted to understand the arrangements used by the vendor to develop knowledge, the participants involved in these arrangements and how these worked together to support the design of DHIS2. The first step was looking through transcriptions from our interviews and further code these by highlighting the text fragments we deemed most relevant. We searched our data for any activity related to how the DHIS2 core team was developing knowledge to support the design of the software and found that the core team had several arrangements that they used for this process. The full picture of knowledge development took time to establish, due to the complexity of intertwined actors and arrangements used.

With more interviews and coding we slowly built up a long list of arrangements that the DHIS2 core team was actively using. In making this list, we also found that other participants than core team members were partaking in the knowledge development process and our next goal became to map out the arrangements and the participants connected. Image 3-3 illustrates the further categorisation on a whiteboard where we mapped out who was involved with what arrangements.

Image 3-3: Network of participants and arrangements

As shown in image 3-3, we found that in the current arrangements for knowledge development, the vendor (named product team in image 3-3) had several arrangements where they were not directly involved with the user organisations and implementations partners (named global community in image 3-3). We saw that the other actors were often placed between the core team and the field and or that they were monitoring digital communication channels instead of talking to the user organisations directly. This mapping helped us understand the broader network of participants and arrangements that became a basis for further analysis. While this insight was useful, understanding how it all worked together in the process of developing knowledge to support the design of DHIS2 was still unclear and further mapping was thus needed.

To better understand how knowledge was developed to support design of DHIS2, we focused on mapping out how knowledge about user organisations was traveling through the network of participants and arrangements. This was done together with one of our informants and was later shown to other informants that confirmed, corrected, or added to our understanding. Image 3-4 illustrates one of the early iterations of how our perceived understanding. This map gave an altogether view on the current arrangements for developing knowledge, the participants involved in the process, and the relation between them in the process of supporting the design of DHIS2. Essentially it gave insight on how the vendor developed knowledge to support the process of designing DHIS2 as it is today.

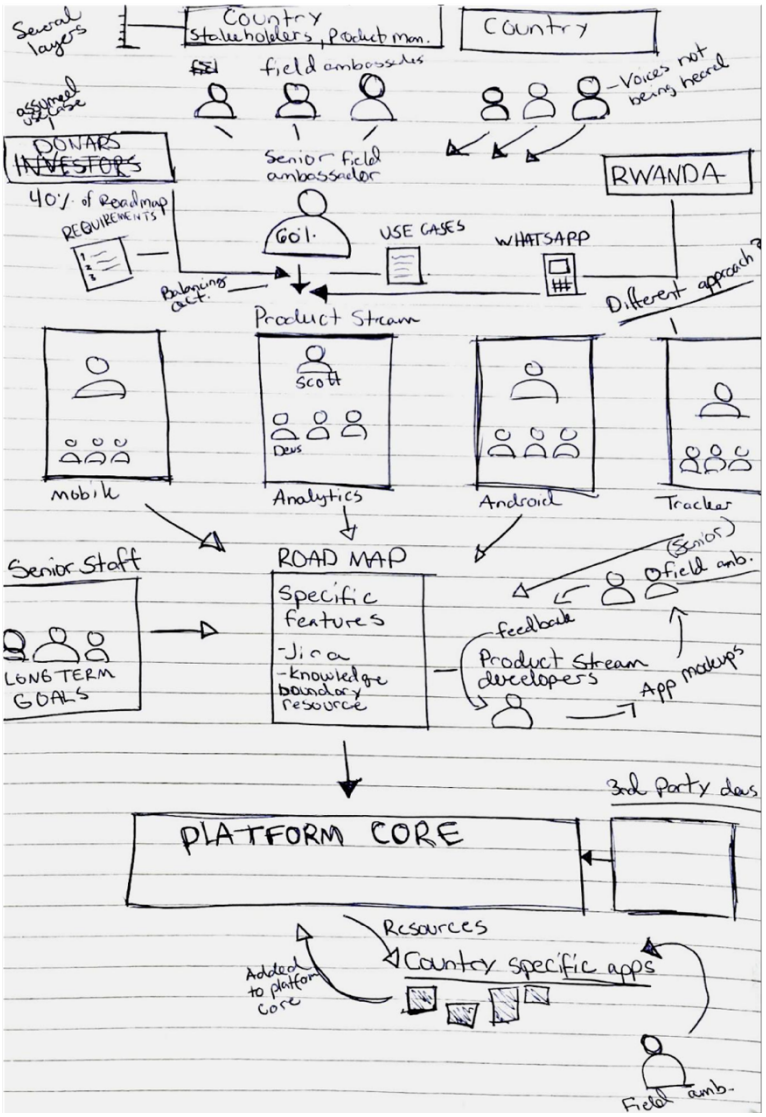


Image 3-4: Conceptual model of how knowledge travels within the ecosystem

3.4.1.2 Research paper in the IRIS conference

To support our analysis, a literature study on the design of ES was conducted in parallel. We found work on generification by Pollock et al. (2007) especially relevant as we saw similar traits and examples in our study. We further used this as a lens for grouping the arrangements we had mapped into three three “approaches" that each contributed to develop knowledge to support the design of DHIS2. This analysis became the basis for a conference paper (see appendix 1) we submitted to the IRIS 2021 conference for discussion purposes. The research question we addressed in this paper was *“How can a vendor gain a coherent understanding of a diverse user base for informing generic design?*”. Image 3-5 shows the three “approaches" we presented as our contribution in the paper.

Primary role in informing design	**Approach to inform generic design**	**Examples of arrangements**
Align user needs	Practices inform design through a process generification work	Outsourcing alignment work to expert user Aligning needs in global ticketing system
Gain insights into specific practices and needs	Direct engagement with user organisations, gathering more granular needs directly from the context in which software is implemented	Gathering requirements directly from a group of user organisations
Gain overview of the needs of the broader community	Providing global communication channels and monitoring them to gain perspective on diversity within user groups	Monitoring global communication channels

Image 3-5: The three approaches from our research paper

Through sessions of discussion at the conference, we understood that the major weakness of the paper was that it did not provide enough depth into the findings to warrant new knowledge. From this process, we landed on three avenues that would further our research. Firstly, there was a need to look broader than the literature on generification to see if other notions could be used to elaborate and better explain the process of developing knowledge for supporting the process of designing ES. Secondly, further data collection and analysis

was needed to provide more insight into how the process of developing knowledge occurs. Lastly, we realised the need for looking deeper into the aspects that underpin our identified approaches to support design, and how they have evolved over time.

3.4.2 Forming our contribution

As noted, during the problem formulation phase we discovered that the core team desired for a greater overview over the arrangements they used for developing knowledge to support the design of DHIS2 as well as potential ways to improve these arrangements. This was still in focus when continuing our work. The feedback gained from the IRIS conference gave insight to how we could better contribute to solving this problem by increasing the emphasis on gathering more empirical material. Following, we conducted more interviews followed by another round of thematic analysis (see image 3-6). First we identified various codes in our newly acquired data, then we categorised the codes into themes that we found relevant based on our topic of interest; Developing knowledge to support design of DHIS2. We identified four themes that were recurring: 1) challenges with developing knowledge to support design, 2) evolutionary dynamics of the DHIS2 ecosystem, 3) the arrangements for developing knowledge and 4) The core team's ideological focus. The latter was later deemed irrelevant.

Image 3-6: Thematic analysis from our second round of interviews

3.4.2.1 Constructing a timeline

We wanted to further investigate the relation between the challenges, arrangements and how these have evolved, as this could further our understanding of how the DHIS2 core team develops knowledge. In order to map out the relation between the identified themes, we constructed a timeline of events, focusing on how the arrangements for developing knowledge to support the design of DHIS2 had evolved over time. As stated in the background, many changes have been made to how the DHIS2 core team develops knowledge to support their design process since DHIS2' origin in 1994. We were interested to see how arrangements and challenges evolved with the DHIS2 ecosystem increasing in number of user organisations and implementations. By constructing the timeline, we mapped out how arrangements were introduced or adapted as a response to the challenges the core team had encountered. Image 3-7 illustrates how the timeline of events was mapped out, highlighting the challenges and responses. Analysing the empirical data by mapping out the timeline illustrated how challenges that occurred were often related to the evolutionary dynamics of the DHIS2 ecosystem. In essence, as the software extended and scaled globally, new challenges occurred which triggered the need to introduce new arrangements, or changes to current arrangements.

Challenge →	**Response** →	**Challenge** →	**Response**
Previous arrangements placing the DHIS2 core team several levels above the user organisations. Communication barriers leading to potential valuable information being lost	Introducing global ticketing system. Allowing anyone to within the ecosystem to request changes, report bug fixes or address the need for new functionality	Global ticketing system leading to overflow of information. Many request for changes too specific to be catered for in the ES.	Introducing a public roadmap of planned features. Making user organisations aware of what types of features will be prioritised. Illustrating that only needs of broader relevance will be catered for.

Image 3-7: Timeline of challenges and responses

The findings from mapping out the timeline of how arrangements had evolved made us turn to information infrastructure theory (Hanseth & Lyytinen, 2010). Specifically we found the notion of cultivation to be very descriptive to how the core team constantly works to evolve their set of arrangements towards supporting the design of DHIS2. Further looking into literature on both software ecosystems and design of ES, we found that Johnson et al. (2014) had coined the term knowledge infrastructure to describe the combination of arrangements used by a vendor for developing knowledge to support the process of designing ES. We found this concept useful in capturing the set of arrangements we had discovered in our empirical findings. Both cultivation and knowledge infrastructures became a part of our theoretical lens and used for further analysis.

At this stage in the process, we had identified in the following in our empirical data: 1) challenges relating to developing knowledge to support the design of DHIS2, 2) arrangements responding to those challenges, and 3) how these have changed over time, subject to the evolutionary dynamics of the DHIS2 ecosystem. By highlighting the relation between these findings, the notion that the DHIS2 core team was not only working reactively when altering their combination of arrangements, but also working proactively based on a set of key areas of focus, emerged. The focus henceforth was then to contribute with considerations for how a vendor proactively can cultivate a knowledge infrastructure to support the design of an ES.

From the literature, we had established a theoretical framework stemming from ES within software ecosystems and used cultivation from infrastructure theory and knowledge infrastructures to describe the arrangements and their evolving nature. This theoretical framework was used in our final stages of analysing the data, which led to the construction of our contribution.

3.4.2.2 Constructing our contribution

In the final stages of analysing our data, we did a theory driven analysis (Braun & Clarke, 2006) centered around investigating considerations for a vendor when cultivating their knowledge infrastructure. Image 3-8 illustrates the process of the theory driven anaylsis. The column to the far left (titled activities in image 3-8) shows every arrangement used by the vendor. The next column titled "goal" shows the intended goal of each arrangement. We saw that several of the arrangements shared similar goals and a table was created where different arrangements were placed under four broader themes (under the column named approaches in the table in image 3-8).

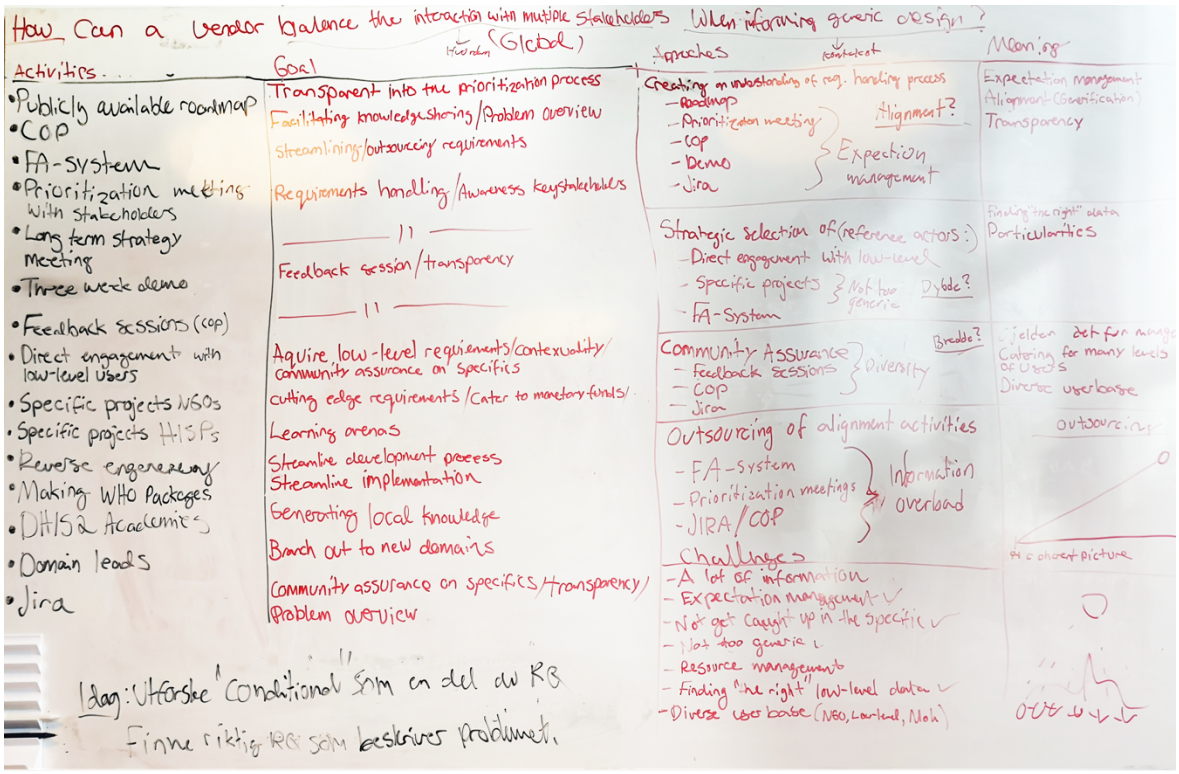


Image 3-8: Theory driven analysis

We then worked iteratively to reorganise the goals behind each arrangement into more general considerations. This was done by revisiting the empirical data for each arrangement, generating codes relating to the goals, and identifying patterns in the overarching themes. After several iterations, the considerations emerged describing what the vendor works towards when introducing new arrangements, or changing current arrangements for developing knowledge. Table 3-2 exemplifies how we went from empirical data, to codes identifying the goals related to specific arrangements and theme highlighting the consideration.

Empirical quote	Code: Goal	Related arrangement	Theme: Considerations
First of all, HISP Rwanda is very organised. Rwanda as a country has a lot of qualities that we would like to reinforce with other nations as well".	Strategic choice of user organisation	Direct stream of communication with a health facility in Rwanda	**Forming strategic partnerships with user organisations:** The vendor forming close relationships with a few strategically chosen user organisations, acting as points of reference for understanding specific work practices and testing usability of the generic software.
"The Rwanda use case has really helped us understand the clinical workflow. We always get a general sense of how people are moving around the different apps"	Reference point for understanding work practice		
"In order to get quick feedback on these new features, we have identified places which can implement new things quickly and become pilot testers, as a lot of DHIS2 implementations are 1 or 2 years behind the last release due to limited technology to update the software."	Strategically chosen user organisations to provide feedback, points of reference for further development	Testing developed functionality with selected user organisations	

Table 3-2: Identifying codes and themes to form a consideration

3.5 Chapter summary

In summary, the empirical research in this thesis is based on a engaged research project in collaboration with the DHIS2 core team, where the aim has been to address a real-world problem situation while contributing to academic literature (Mathiassen & Nielsen, 2008; Mathiassen, 2017). The engaged research project has been conducted in an iterative manner. The three activities of scoping the project in collaboration with the DHIS2 core team, formulating our research problem and evaluating the form of inquiry used to address this problem has been continuously revisited as the project has evolved. Based on a continuous assessment of feasibility and relevance to address our evolving research problem, case study has persisted as the form of inquiry guiding our research project. We have utilised several methods for data collection, focusing on gathering a broad set of perspectives within the DHIS2 core team through interviews, observations and document analysis. Through these efforts, we have gradually built an understanding of how the DHIS2 core teams works to develop knowledge about their diverse set of user organisations, and how this has evolved over time. The character of our case study can be classified as interpretive, attempting to understand the context of interest through the intersubjective meanings and experiences of the participants that exist within it. The analysis of our empirical data has been a parallel process of engaging in both the literature and empirical data, allowing for an abductive process where our contribution was iteratively shaped by both. This has been done through several rounds of thematic analysis, and by developing a theoretical framework to describe the phenomenon identified. The end result of this process has been our contribution of four considerations for cultivating a knowledge infrastructure. These considerations are described and discussed in chapter 5. In the following chapter, we will describe the findings from the described research process.

4 Findings

In this chapter, we will describe how DHIS2 has evolved from a one-country implementation to a global ES and look at how the knowledge infrastructure has been cultivated over time. We start by giving a backstory of how DHIS2 came to be, along with the few initial arrangements for developing knowledge to support the design of the DHIS2 software. Further, we look at the challenges the DHIS2 core team encountered as the software ecosystem expanded globally, along with the introduction of new arrangements to overcome these challenges. Finally, we summarise all of the arrangements that we later analyse and discuss to identify four considerations for vendors when cultivating a knowledge infrastructure. We have divided the evolution of DHIS2 and the knowledge infrastructure into four phases, summarised in table 4-1. The first two phases describe the origin and the early arrangements and the latter two describe the knowledge infrastructures evolution in recent years.

Phase	Description
Phase 1: The origin of DHIS2 and the Knowledge infrastructure	The initial starting point of DHIS2 and the knowledge infrastructure. To develop knowledge, core team members gathered needs directly from user organisations.
Phase 2: Establishing partner organisations	The core team introduced HISP groups with field representatives. User organisations contacted anyone to have their needs accommodated.
Phase 3: Creating structure and hierarchy	The core team introduced more structure and hierarchy by implementing the field ambassador arrangement. Design- and prioritisation meetings were also introduced.
Phase 4: Reaching a broader set of user organisations	The core team introduces arrangements that focus on making them more available to user organisations as well as introducing direct lines of communication.

Table 4-1: Summary of the four phases of the knowledge infrastructure

1.1 The origin of DHIS2 and the knowledge infrastructure (Phase 1)

4.1.1 Collaborating with specific implementations

DHIS2 started as a collaborative project between the University of Oslo and the University of Cape Town and was initially developed as a stand-alone application for smaller parts of South Africa. The software was first piloted in a few districts before it was released nationwide and became South Africa's national standard as an ES for health management. (Braa & Sahay, 2017). With DHIS2 user organisations only operating within the borders of South Africa, the knowledge infrastructure was also very limited. Initially, the only way to develop knowledge about user organisations was by a few DHIS2 core team members travelling to a given user organisation and directly gathering needs. These were then developed into knowledge to support the design of DHIS2.

With time, the software was gradually adopted into new user organisations as neighbouring countries to South Africa also took DHIS2 into use. The expansion of the software resulted in user organisations outside of South Africa providing other needs than previously attained by the core team. With the increase in the number of user organisations an uncertainty among core team members emerged. As each member had its own experience from the user organisation they had visited, it became increasingly difficult to separate the most eminent needs from less important ones. The lack of accord among core team members was challenging and brought forward a new arrangement that would help them decide what to prioritise. The new arrangement had a parallel data gathering focus, meaning that instead of core team members visiting one specific DHIS2 user organisation, they would now visit multiple to acquire a wider range of needs. These would then be prioritised into a list and become the roadmap for coming features. This helped ease the uncertainty from the previous arrangement as all of the core team members were now aware of what needs to prioritise first, based on the roadmap they had created collectively. Figure 4-1 illustrates how the roadmap came to be.

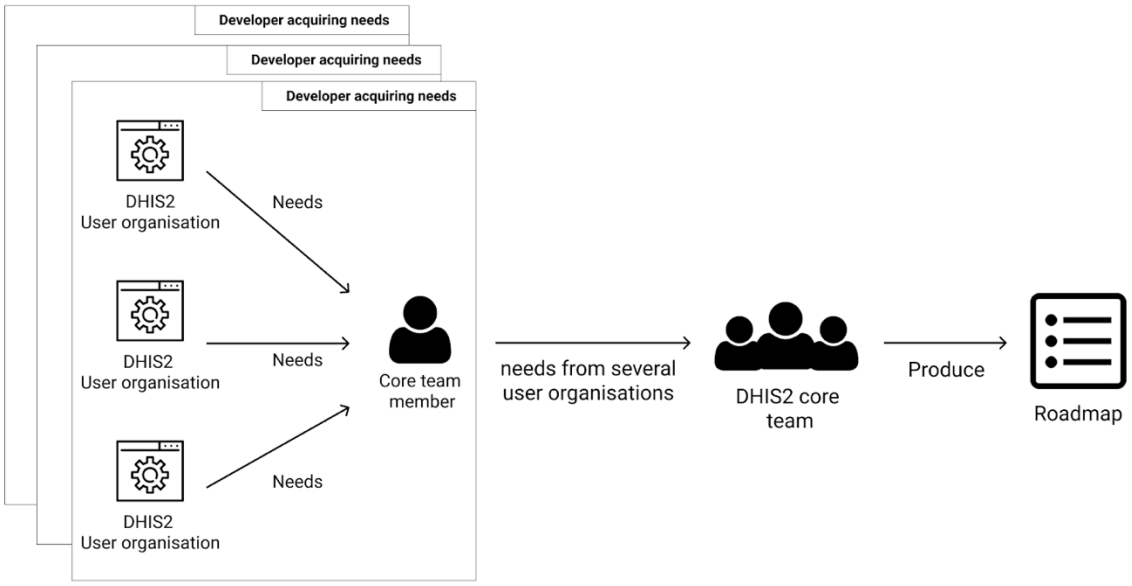


Figure 4-1: Creating the roadmap

For a time, this arrangement worked well but with an increase in software popularity, DHIS2 further expanded beyond the first few countries that had implemented the software. Core team members were travelling to new locations and piloting DHIS2 to new user organisations resulting in the core team no longer having enough resources to be directly involved with all of them. This brought forward a change in the knowledge infrastructure as the current arrangement with direct contact between the core team and the field proved resource-heavy. It was too demanding to develop the necessary knowledge for understanding all of the user organisations. The solution was to employ new arrangements to gather needs more efficiently by utilising implementation partners in the process. The next phase explains how this became a focal point for the vendor when developing knowledge. Figure 4-2 summarises phase one's arrangements and main challenge.

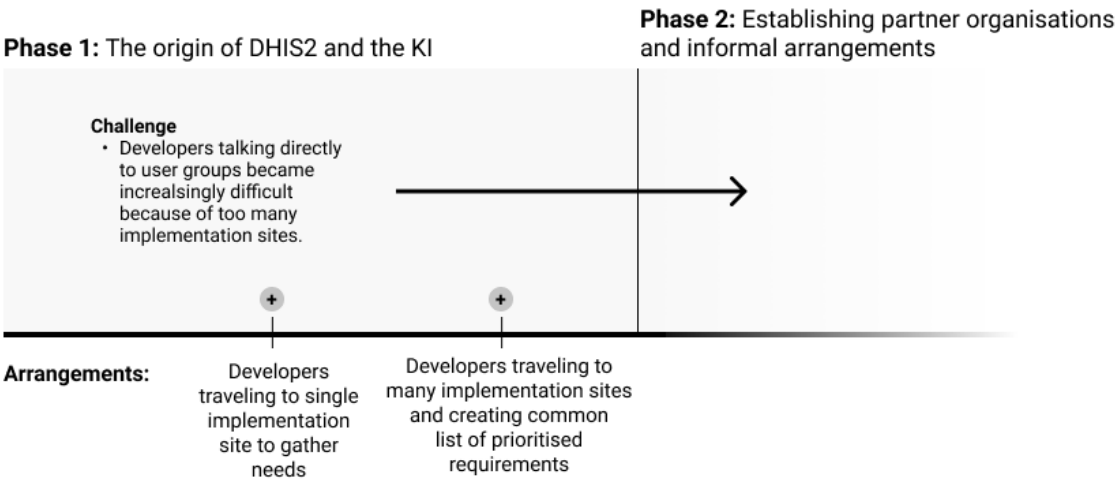


Figure 4-2: Summary of phase 1

4.2 Establishing partner organisations and informal arrangements (Phase 2)

4.2.1 Establishing a network of partner organisations

The rapidly growing number of user organisations became increasingly difficult for the DHIS2 core team to handle. Too few core team members and many new user organisations resulted in needs from several user organisations not being included in the knowledge development process. As a response, a new arrangement was initiated where a network of trusted implementation partners called HISP groups was built up. HISP groups are responsible for providing support on software development, capacity building, and generally strengthening DHIS2 as an ES in given regions of the world. They also feed contextual knowledge about user organisations within their region up to the DHIS2 core team, being geographically closer to the contexts of use. The full network of HISP groups was not established momentarily, but with more user organisations appearing in new regions, a network of implementation partners started to be built. To make the process of feeding knowledge about the user organisations more efficient, each group also appointed a representative who was responsible for supplying the DHIS2 core team with knowledge. One informant described the relationship like this:

"You have groups in Tanzania, one in West Africa, Sri Lanka, India, Vietnam etc. They would feed [needs] into Oslo. It was a fairly informal system, it was based on who you knew. If you had the email of the Dev Lead, you would email him even if it wasn't him that was supposed to be working on it".

The arrangement with appointed representatives resulted in an unstructured and informal system where both the HISP groups as well as any user organisations were pushing their needs to whomever in the core team they had the contact information of. This had a hybrid impact in both a positive and negative manner as it gave access to user organisations who may otherwise be hesitant or be intimidated by layers of hierarchy to get their voices heard, but it also provided disorganisation in which people were prompted with issues they had no control over. In summary, the system became a bottleneck as the DHIS2 core team was overflowed with specific needs from anyone that had their contact information. The core team were not able to distil these various channels of informal communication into useful knowledge that would support the design of DHIS2. This resulted in an ambiguity between core team members as some were developing features without any thorough assessment of the broader relevance and urgency of the feature. One core team member explained the challenge:

"So what we had was a lot of the [core team members] making whatever they wanted or thought was cool. They still had a general idea of what the use case was but they were not necessarily developing specifically off the needs that we were getting. That connection was lost."

"That connection" mentioned here is referring to the connection between the various user organisations and the DHIS2 core team. While the core team had an understanding of more generalised needs, the particularities provided directly by user organisations were no longer accounted for in the development process due to disorganisation and information overflow. The demand for more structure in the arrangements became eminent as valuable information was lost in the overflow of needs coming in. To ensure this, the prior utilisation of HISP Groups with representatives was further developed into a more formalised structure which we described in the next phase. Figure 4-3 illustrates the arrangements for

developing knowledge in phase two and describes the main challenges that the core team encountered.

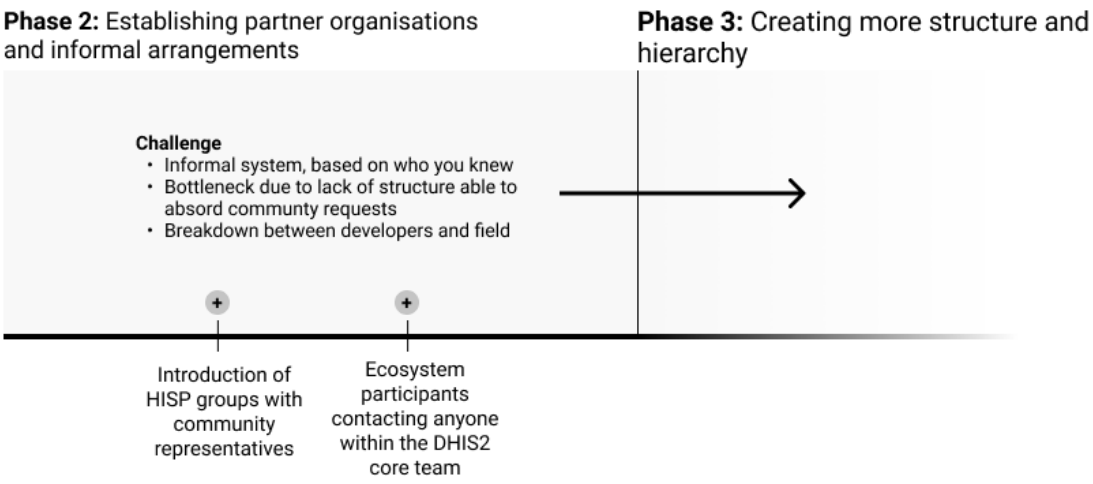


Figure 4-3: Summary of phase 2

4.3 Creating more structure and hierarchy (Phase 3)

4.3.1 Outsourcing the gathering of needs to field ambassadors

With too little structure for developing knowledge to support the design of DHIS2, many user organisations felt that the software no longer attended to their needs. Some expressed that their voices were not being heard in the development process due to a missing connection to the core team. In order to solve this, a new arrangement ensuring that the needs of user organisations were included when developing knowledge, was then introduced. The result was a continuation of the previous HISP group initiative, but in a more structured and efficient manner called “field ambassadors”. Similar to the previous arrangement, this arrangement is based on representatives having an elevated status within a HISP group. The representatives are still in charge of feeding knowledge about user organisations within their region up to the DHIS2 core team. Additionally, this new arrangement makes the field ambassadors also in charge of prioritising and filtering knowledge about the user organisations. This had to be provided to the core team as a list of needs, ranking from most important to least important. The new arrangement helped solve the previous challenge of information overflow as it brought more structure to the

process of developing knowledge. Instead of user organisations directly providing needs to anyone they knew within the core team, needs now had to go through the field ambassadors. Knowledge about user organisations was thus better included in the process of developing knowledge as the core team was only fed with relevant prioritised lists, instead of many irrelevant issues as previous. A core team member explained the process in further detail, which we have illustrated in figure 4-4:

"They come up with a prioritised list of [needs]. I then have communication with all of them to break down these priorities which are not usually feature specific, into specific features. Then we [the core team] develop mockups and feed the mockups back to them and create an agile process of refining the features and user stories before we feed it into our dev-team."

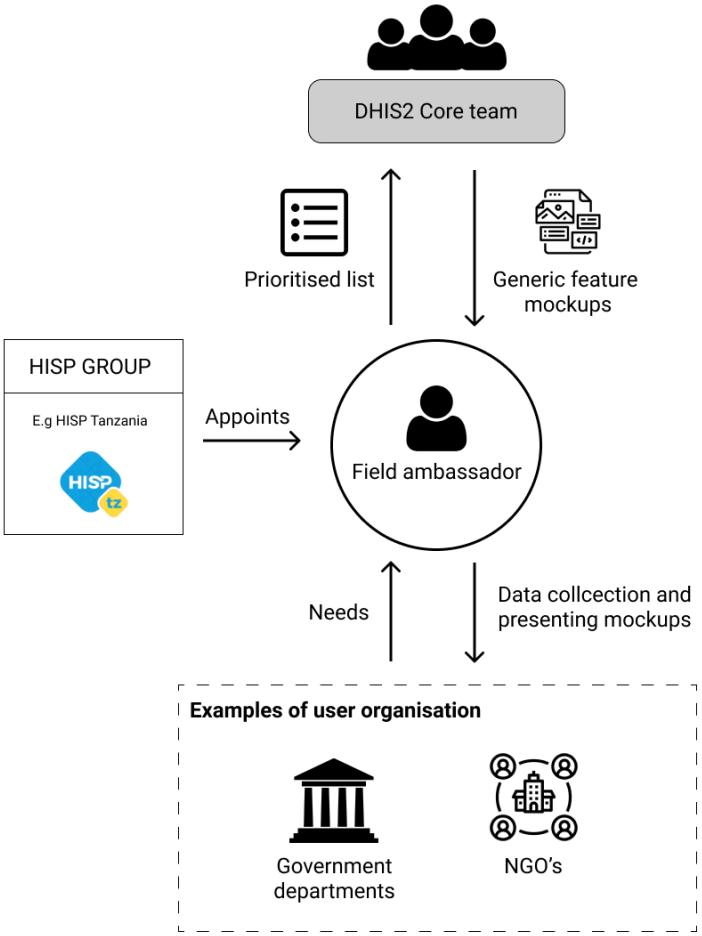


Figure 4-4: The field ambassador arrangement

In general, the arrangement affords for various user organisations to be represented in the development process in a much more cost-effective manner and the back and forth communication ensures that the design of DHIS2 is grounded in needs from various user organisations. While this had a positive impact, a challenge to this arrangement relates to the core team no longer being in control over the type of mindset that is put into prioritising the needs. From the core team perspective, needs should be prioritised based on their broader relevance to other user organisations. Accordingly, the needs relevant for the greatest number of user organisations should then be prioritised first. The core team wants the field ambassador to have the same focus in mind when they produce their lists of ranked needs from their given region. To solve this challenge, they try to get the field ambassadors into the core teams' way of thinking:

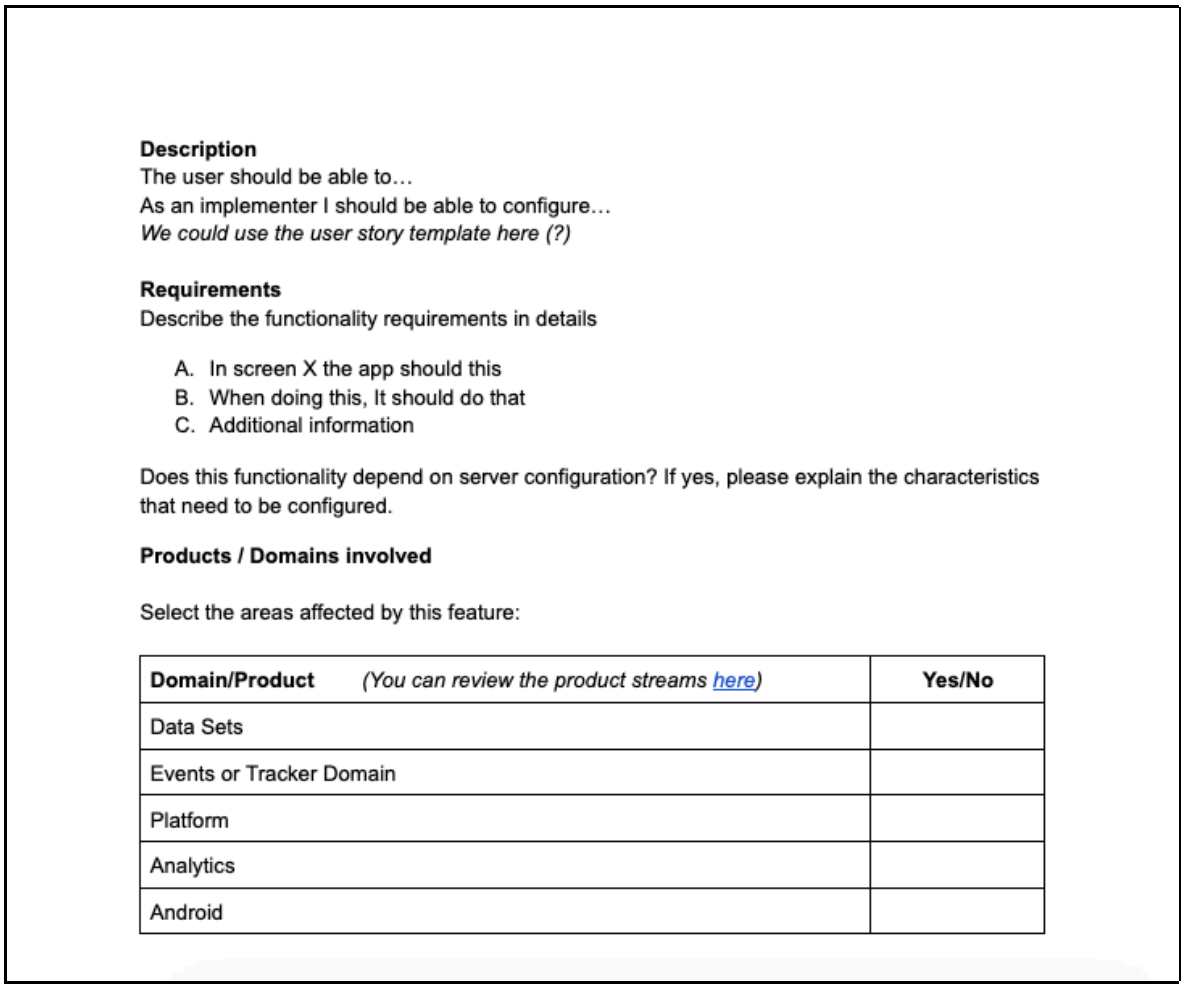
"We actually do kind of want to empower them to have more of a role and decide what is the best way to capture this information. "How do we think of this generically?" "How do we try to align and separate requests from multiple countries into one functional request?" We want them to get them more involved in our way of thinking in terms of making things generic.."

"*Our way of thinking in terms of making things generic*" in this sense means that they need to look beyond the context in which they are gathering needs from and try to see it in relevance to the broader ecosystem. The next two arrangements further describe ways in which the core team enforces this way of thinking by adding more control over how the field ambassadors conduct their work.

4.3.2 Ensuring unity in outsourced work

As field ambassadors are now responsible for developing a major part of the knowledge about user organisations and feeding the core team with a prioritised list of needs, the DHIS2 core team wants the field ambassadors to have a focus on prioritising needs relevant to many. In essence, prioritising specific needs that are too contextual, will neither benefit the user organisations, the field ambassador nor the vendor as it results in DHIS2 becoming less relevant and usable to most of the user organisation. To avoid this transgression, the vendor has introduced a specific data form that the field ambassadors are to use when formulating the needs of user organisations (see image 4-1). It requires the field ambassador

to fill in a description of the need, information about who it is for and what part of the generic software it involves.



Description
The user should be able to…
As an implementer I should be able to configure…
We could use the user story template here (?)

Requirements
Describe the functionality requirements in details

A. In screen X the app should this
B. When doing this, It should do that
C. Additional information

Does this functionality depend on server configuration? If yes, please explain the characteristics that need to be configured.

Products / Domains involved

Select the areas affected by this feature:

Domain/Product *(You can review the product streams here)*	**Yes/No**
Data Sets	
Events or Tracker Domain	
Platform	
Analytics	
Android	

Image 4-1: Form for formulating needs

This standardised way of formulating needs ensures two things: One, it is a control mechanism that ensures needs received are coming from user organisations. The goal of implementing the field ambassador system was to develop knowledge from user organisations that would otherwise be difficult, due to a lack of resources. Specifically, clinical workers such as nurses and doctors are at times neglected in the design process because of their limited capacity to bring forward their needs. With this form, the DHIS2 core team has a larger degree of control over where needs are coming from, making the prioritisation of these types of user organisations a less demanding task. Two: the form is a

way of enforcing less contextuality in the needs acquired as they have to be presented in a generic way that is of relevance to others. The form asks specifically to describe the wanted functionality in detail and the core team will neglect needs that are described as context-specific problems or ones that are not actionable towards the design of DHIS2. The challenge is to make this mindset clear to the field ambassadors prioritising and providing the list of needs. One way of creating awareness is through the various meetings occurring between the DHIS2 core team, implementation partners and user organisations.

4.3.3 Design and prioritisation meetings

The new arrangements for developing knowledge about user organisations are more efficient but also lead to challenges. While the form ensures that needs on the list are actionable towards the design of ES, it does not promote the core team's desired way of prioritising the needs. Specifically, the focus of prioritising needs that affect the largest number of user organisations is not always present, as some field ambassadors prioritise otherwise e.g emphasising one specific user organisation. To ensure that all of the knowledge coming in from the field ambassadors is similarly prioritised, the DHIS2 core team needed to introduce other arrangements. One solution was to incorporate two types of meetings that serve different roles in ensuring unity in the process of developing knowledge.

4.3.3.1 Design meetings

The most frequent meetings are the “design meetings” where HISP groups and user organisations are weekly invited to discuss and give feedback on features that are in development as well as submit needs that may be of relevance to those features. A general challenge is to make both HISP groups and user organisations aware of the generic focus the DHIS2 has when prioritising needs. Design meetings are used to solve this challenge as it affords for core team members to explain and discuss why some types of needs are prioritised and why some needs are neglected. A DHIS2 core team member describes it as building a culture:

"When they [field ambassadors] send in a request, they understand that it needs to be a generic request and that they need to present it in a generic way. And I think that kinda just the culture we built around it, and they kind of understand that to a larger extent."

The meetings are a formalised way of ensuring the features coming in are of the type that is useful to a broader set of user organisations. The meetings are also a discussion forum where DHIS2 core team members and field ambassadors together can work out and reformulate needs more generically. Oftentimes, needs can be obscurely stated or have to be translated into a format that is actionable towards the design of DHIS2. These meetings provide a space where the people in charge of designing DHIS2 and the people in charge of providing the knowledge to support this process, can come together to establish the same mindset. While these weekly design meetings ensure that the features in development are of the right type, agreeing upon the long term focus is also an essential to establishing a similar mindset.

4.3.3.2 Prioritisation meetings

Prioritisation meetings are biannual meetings where representatives from HISP groups and user organisations are invited to participate in presenting their needs as well as vote on what needs they think should be prioritised. The long-term focus of DHIS2 development is a democratic decision made by the meeting participants, where the needs that get the most votes, will also be prioritised first. Since each HISP group and user organisation may provide multifold needs, it is necessary to limit the number of needs that are being discussed in the meetings. Initially, the workflow of these meetings was to allow every group attending to present as many needs as they sought fit, but this proved inefficient due to extensive amounts of needs at each meeting. A DHIS2 core team member explained it like this:

"Earlier we tried to pull up all the needs and discuss them, but it proved impossible with 40+ people trying to go through hundreds of needs. Even with a couple of days of meetings we weren't getting there. Next time around we had people submit their top 5 requests based on their own experience."

These meetings were originally ineffective, but with time proved important as a way of ensuring unity among HISP groups and user organisations. With each participant

presenting their top five needs, other participants of the meeting would see how common or uncommon their own needs are in relation to those presented. In the later process of agreeing upon the long term goals for the DHIS2 software together, HISP groups and user organisations were made aware that needs that are relevant for many will be prioritised over those who only affect the few. This mindset is important to make needs coming in as actionable towards the design of the DHIS2 as possible.

In summary, this meeting is an essential part of the knowledge infrastructure as it provides the DHIS2 core team with the most relevant needs from every stakeholder as well as a way to show user organisations that their specific needs might not be relevant to the broader ecosystem and thus will not be prioritised. While the meetings and the hierarchy from field ambassadors created more structure, it made it increasingly difficult for the core team to reach the user organisations directly. Contrariwise, the core team also became unavailable to the user organisations as layers of field ambassadors have been placed between them. It is desirable for both the core team and the user organisation to not only be connected through field ambassadors, as direct forms of communication give a more detailed insight which may be useful when developing knowledge to support the design of DHIS2. To solve this challenge, further arrangements were needed to make the core team more available to user organisations which we describe in the last phase. Figure 4-5 illustrates the arrangements introduced in phase three and summarises the challenge that highlights the need for more arrangements in phase four.

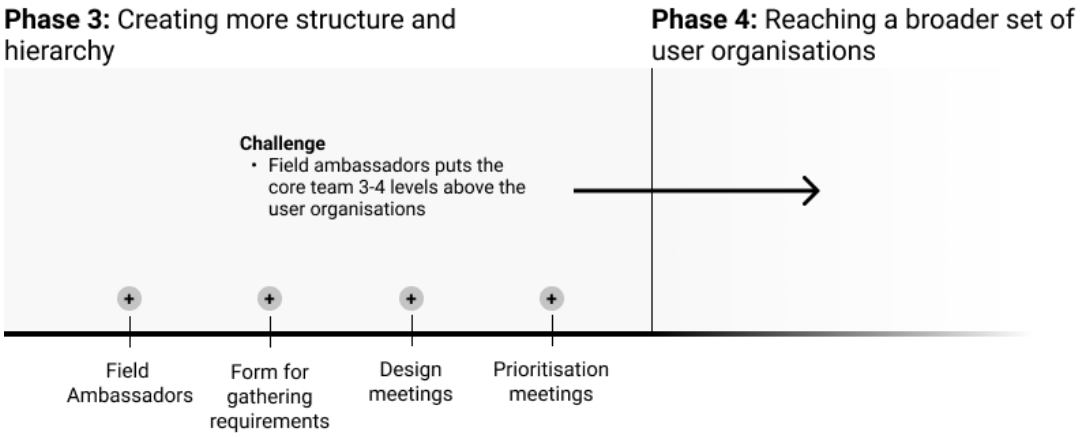


Figure 4-5: Summary of phase 3

4.4 Reaching a broader set of user organisations (Phase 4)

4.4.1 Global ticketing system - Jira

The hierarchy created by the field ambassador system imposes a challenge as it is limited by the number of user organisations the field ambassadors confer with. Each field ambassador may strategically choose its sources of information and at times, there is uncertainty as to what premise the list of needs going to the core team is prioritised on. To solve this, the core team introduced Jira, a global ticketing system that allows any participant within the ecosystem to request changes, report bug fixes or address the need for new functionality. The system is split up into different ongoing projects, where each project has its backlog of needs. Image 4-2 illustrates the backlog for a part of the DHIS2 software solution, and image 4-3 a typical bug report.

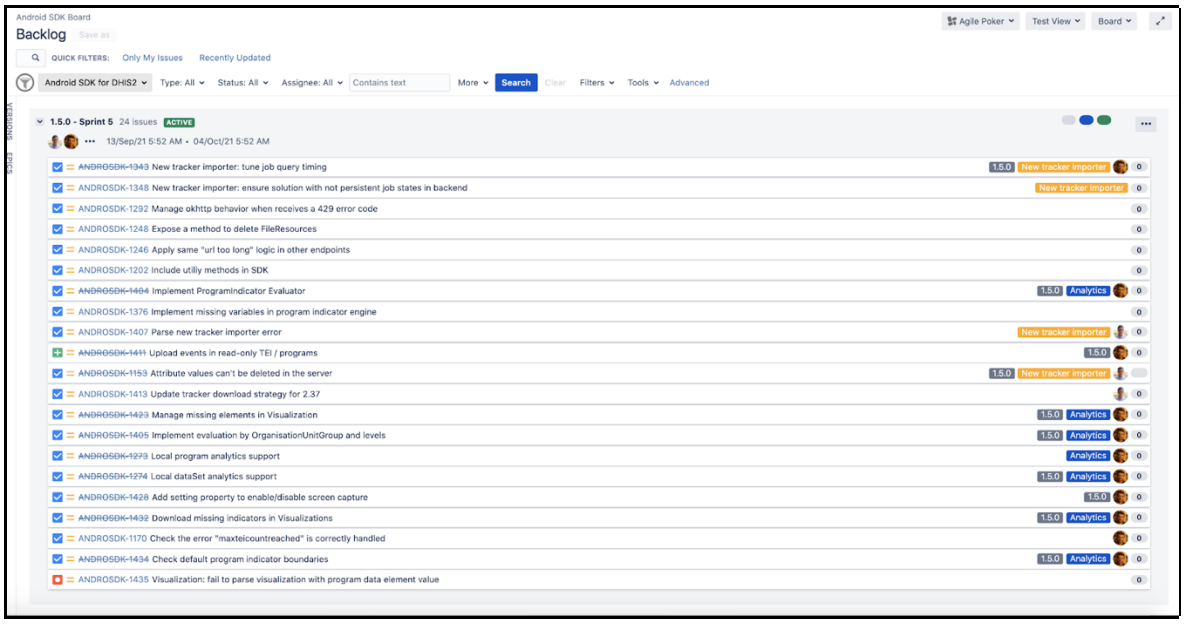


Image 4-2: Backlog for a part of DHIS2

The openness of Jira enabled the core team to gather needs directly from a broad set of user organisations that was previously only possible by travelling directly to them or through levels of hierarchy with field ambassadors. With such a limitless system, challenges relating

to an overflow of needs is imminent. One DHIS2 core team member expresses some concern:

"There is more in Jira that anyone could do in a lifetime and there is a huge backlog of things we haven't prioritised. But we have multiple efforts that can lead us to the right places in Jira. For example, we already have a long-term strategy we are adhering to, we do periodic reviews with HISPs [referring to design meetings], and we have a roadmap of high-level functionality that we want to meet. And then we can take that back to Jira and look for things related to that and try to bring that into the process."

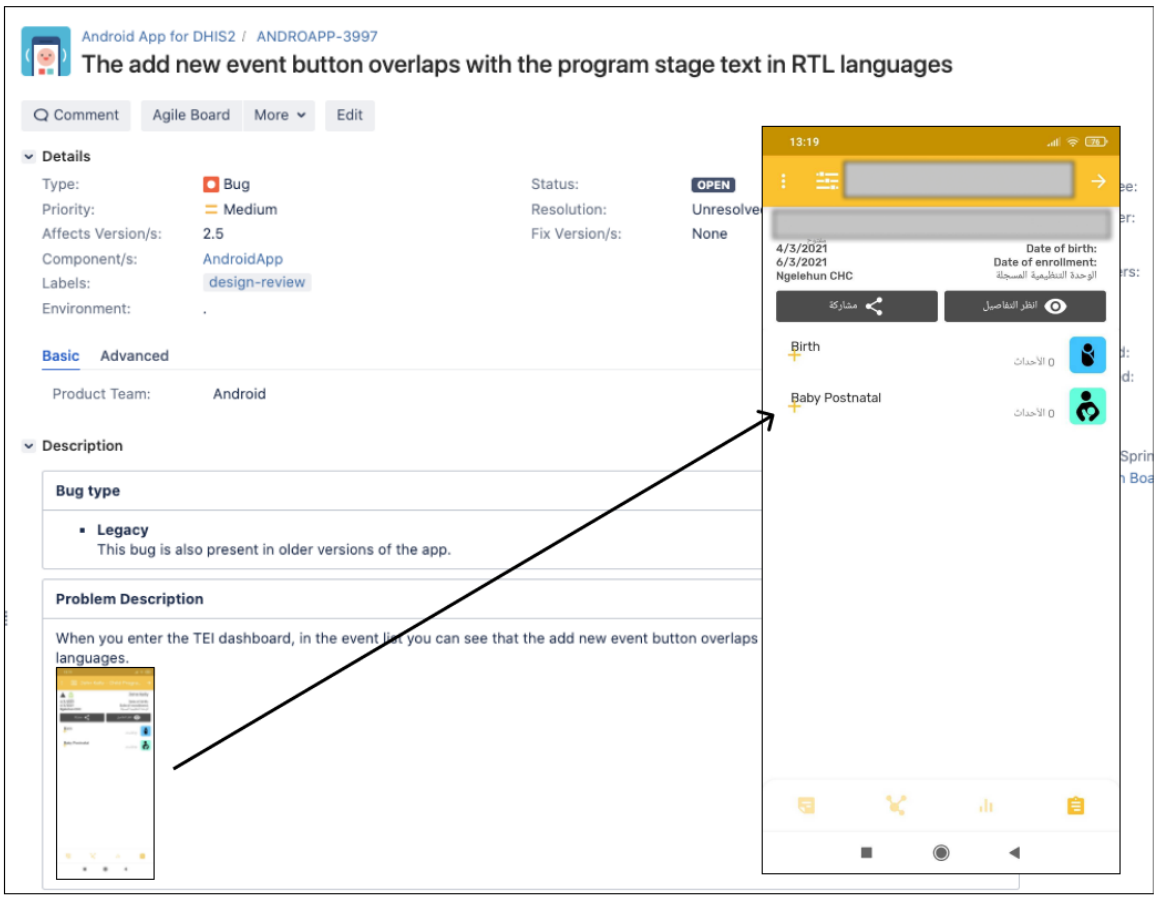


Image 4-3: Bug report

With the accessibility Jira affords, it becomes impossible to go through every need submitted. Essentially Jira is instead used as a way of confirming the usefulness of features

that are in development. Members of the core team can search up submitted needs relating to a given feature to acquire community assurance in whether this need is useful to the DHIS2 ecosystem or not. Jira has also made the process of aligning needs more transparent as user organisations now become aware of what features are being worked on, what features are being prioritised and features that have already been requested.

4.4.2 Global communication channel - COP

Around the same time Jira was added, another digital system was introduced to the DHIS2 knowledge infrastructure. The community of practice (COP) is a globally available website, where HISP groups, members of the DHIS2 core team and user organisations engage in different topics of discussion. The website is centred around user support, capacity building and resource sharing and is a useful digital meeting point. Image 4-4 shows how it is structured in different categories and how participants on the COP can easily join a discussion or create a new post.

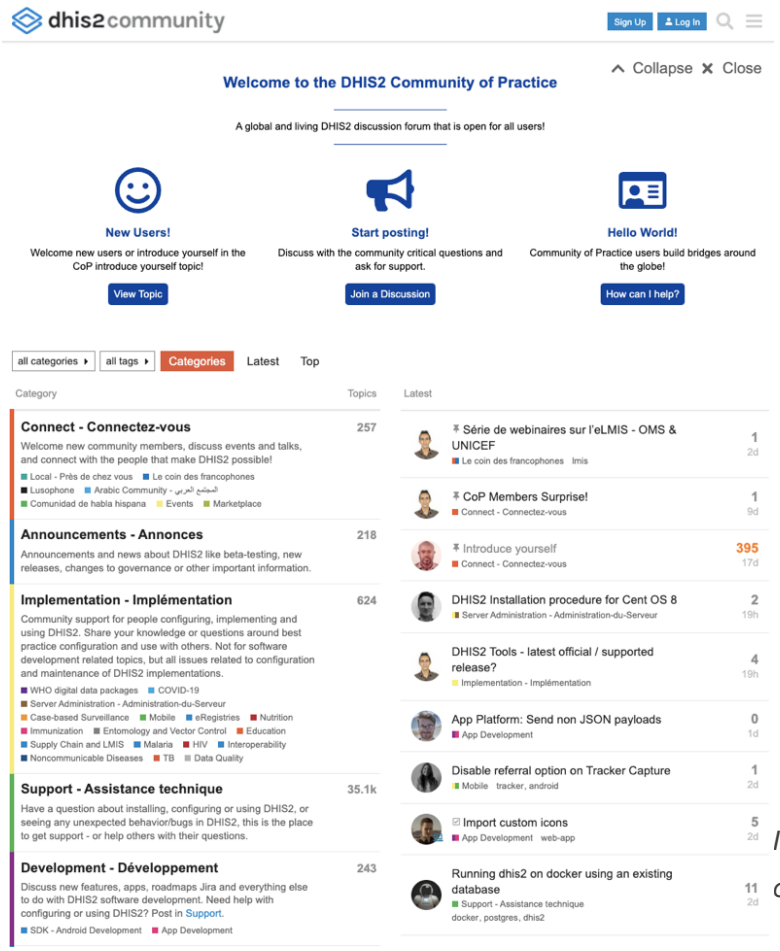


Image 4-4: The community of practice

This arrangement is generally useful to the core team as they can gather information on topics that elsewhere would be hard to acquire. As one informant described it: *"The COP is the most raw, uncurated information source. It gives us a real perspective and understanding of the diversity in the global community"*. This substantiates how it in similarity to Jira removes the barriers between the DHIS2 core team and the user organisations by giving an unfiltered perspective of the ecosystem. While Jira is centred around needs submitted by user organisations, the COP is more focused on sharing knowledge and giving feedback. One example is image 4-5, which shows how it allows the core team to develop knowledge about multiple user organisations on cases that have to be dealt with quickly. Similar to how Jira is used, COP is an arrangement that is used for community assurance by the DHIS2 core team. This is mainly happening through the support channels on the COP or through discussions on features that have been or are to be released. One core team member elaborated:

"There is a community assurance perspective mainly on the support channel where we have a lot of people asking questions. We will develop something and then someone will ask for it. We can then say "we already made this" or "this is coming" and this is reassuring me that we are doing the right thing. It's one more voice in the choir of deciding what we should be doing."

Feedback request on 2.35 dashboard print feature

Development - Développement

CORE TEAM MEMBER [redacted] 9 Feb

Hello all, the analytics dev team is looking for feedback on the dashboard print and print preview feature that were introduced in 2.35. This was highly requested, and enables for dashboards to essentially server as a standard report that can be printed or saved as a pdf that can be shared. Has anyone had any experience using it? Any feedback? Do you know of people using it that we can talk to?

created [redacted] last reply [redacted] 2 replies 102 views 2 users 3 likes 1 link [redacted]

IMPLEMENTATION PARTNER [redacted] 10 Feb

Hi [redacted]

Thanks for sharing, I thought we had shared with you our 2.35 testing feedback

[redacted]

See Dashboard printing feedback.

Domain	Feature	Test comments	Recommendations
ANALYTICS FEATURES	**Dashboard printing**: You can now easily print a dashboard, either with a single item per page or in the preset dashboard layout. This enables you to create a dashboard as a standard printed report for routine meetings or plannings. You can also save a dashboard as a PDF to be shared via email or messaging.	• Awesome • This is good can we have the selected dashboard mailed to a user group • Printer function is a great idea for dashboard usage • All basic print functions are satisfactory (Tested using Google Chrome) • Allow selecting and printing one graph/chart directly from the dashboard	• First page could be improved or removed, guess it's for Dashboard title and description, • The pivot table is being cut if scrollable • Maintain colour in the pivot table in all browsers - Microsoft Edge maintains the Pivot table • Make "More" option available for each dashboard item • Make Print functions more flexible by allowing user to select only the reports they want to print at once, not all the reports in that dashboard • Enable scheduled Emailing of the Dashboard (Dashboard Push to emails) • Embedding the dashboard with other systems.

Image 4-5: The core team receiving feedback from implementation partner

The community assurance on features in development is important for the core team as it indicates if the development process is on the right track. The COP and Jira both afford for assurance as well as developing knowledge on a broader perspective through unfiltered community input while maintaining a certain "distance" to the user organisations. A core team member explained why this is important:

"What any of these groups want is a closer connection to us, and they want some ability to influence the roadmap in a way that gets them what they want." [...] we have to be careful about how we approach this because we don't want to owe a bunch of groups different things." [...] "We are learning how to scrape off and glean what we need without somehow putting expectations in a place like that they somehow are officially part of our roadmap process."

With previous arrangements, the DHIS2 core team would visit user organisations and create expectations the needs being gathered would be accommodated in the design DHIS2. This became problematic as there was no guarantee that other members of the core team would deem the same needs equally important once compared with needs from other user organisations. With these global communication channels, the DHIS2 core team can develop the same knowledge without directly talking to the user organisations and avoid creating expectations that are beyond the bounds of possibility. While it is important for the core team to gather needs without creating expectations, making the user organisations aware of the type of knowledge that will be used for developing DHIS2 is also central. With the number of user organisations steadily increasing, it becomes increasingly difficult to make user organisations and HISP groups aware that only needs covering many user organisations will be accommodated in the generic software. Another arrangement was thus introduced.

4.4.3 Public roadmap

A consistent challenge mentioned by several core team members relates to making user organisations understand their needs may be too specific to be accommodated in the design of DHIS2. User organisations may not always see how they are one of many that want their needs accommodated in the design process. Specifically, this is challenging within the DHIS2 software ecosystem, where user organisations are globally disparate and oftentimes are not aware of each other's existence. It is then important to manage the expectations among the user organisations and make clear what needs are going to be developed upon. One way of solving this challenge was introducing a digital public roadmap to the knowledge infrastructure. Priorly, the roadmap has purely been used internally within the DHIS2 core team as guidance for future development. Recent efforts have been put into making this globally accessible through a website. The public roadmap website

essentially consists of a timeline displaying the release plan for the upcoming version. Any participant of the DHIS2 ecosystem can go in and see what features will be available for the upcoming release. Image 4-6 displays how the roadmap looks with the features that will be included in one of the generic applications that DHIS2 maintains.

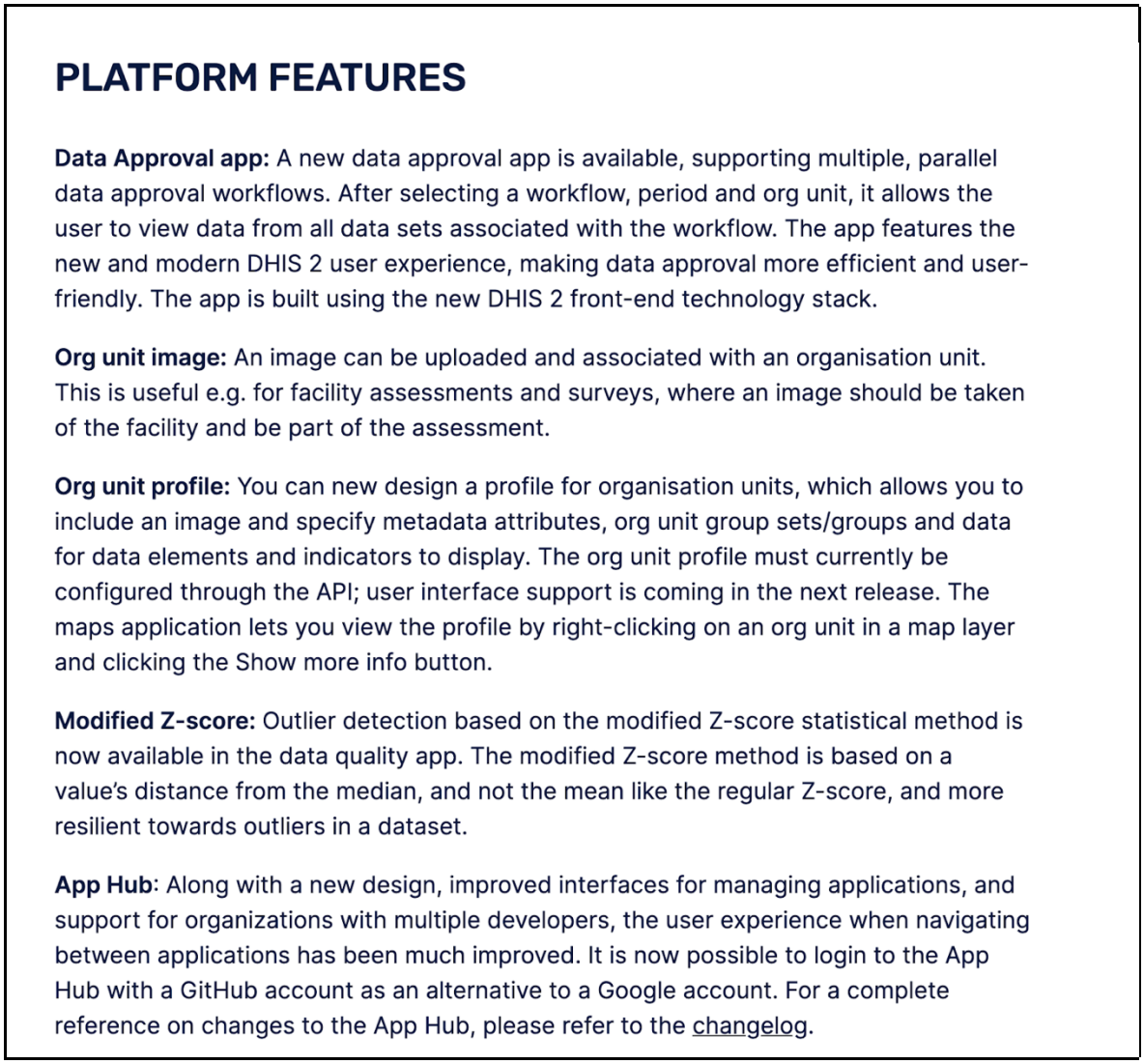


Image 4-6: The public roadmap for one generic application

There are two main reasons for introducing this arrangement: The first reason relates to making HISP groups and user organisations aware of what is planned for in terms of features in the upcoming release of DHIS2. With the core team indicating what features

they are including in the next DHIS2 release, user organisations will see that only features relevant to many will be accommodated, while specific features relevant to only a few will not. This results in more of the needs coming from various user organisations being less specific, making the job of filtering and removing needs that are of less value substantially smaller. The second reason relates to the next arrangement put in place: allowing third-party developers to develop on top of the generic solution and cover functionality not dealt with by the generic software. To do so, the third-party developers need to know what is becoming a part of the core functionality to justify when to develop. The next section will elaborate on how including third-party applications are used as an arrangement for developing knowledge.

4.4.4 Third-party development within the ecosystem

DHIS2's generic capabilities are not capable of covering every need that occurs among its user organisations. Therefore, the core team has also made resources available to develop custom apps that may cover areas where generic functionality does not extend. These applications often solve issues related to very specific use cases, but at times they may also solve issues that apply to a broader set of user organisations. A part of the knowledge infrastructure relates to monitoring such apps and making them generally applicable to include them as a part of the generic software. An informant explained how:

"We are reverse-engineering innovations that have already happened within the DHIS2 ecosystem. Someone would come up with a really cool analytics app, but maybe it is doing too much client-side processing or maybe it works really well in Malawi specifically where there are only 3000 health facilities. But when it goes to scale in Bangladesh, where you have 6000 health facilities, the app starts to crash. So a lot of what we are doing is saying "can we introduce some of the great functionality from the app in Malawi, into the core in a more performance stable way so it can be used globally?""

Many of these custom applications have a limited lifetime, as the third-party developers who built them may have limited resources to maintain them over time. One solution that may extend a custom application's lifetime is by adding it to the DHIS2 App Hub. The App Hub is equivalent to the "App Store" or "Google Play" and consists of apps made by various third-party developers. Here it may become visible to core team members and if deemed

useful, it may become a part of the generic solution and thus responsibility for maintenance is shifted to the core team. One core team member explained the importance of monitoring this channel, saying *"Most of the apps developed we have no idea of it happening, we only find out about them when someone posts them to the App Hub"*. As these third party innovations often represent enhancements to core functionality that are of importance to more than a few user organisations, they might provide inspiration or knowledge that can be extracted to further support the process of designing DHIS2. An example of this happening is the custom app made by HISP Tanzania for extending the dashboard functionality already existing in DHIS2 (see image 4-7).

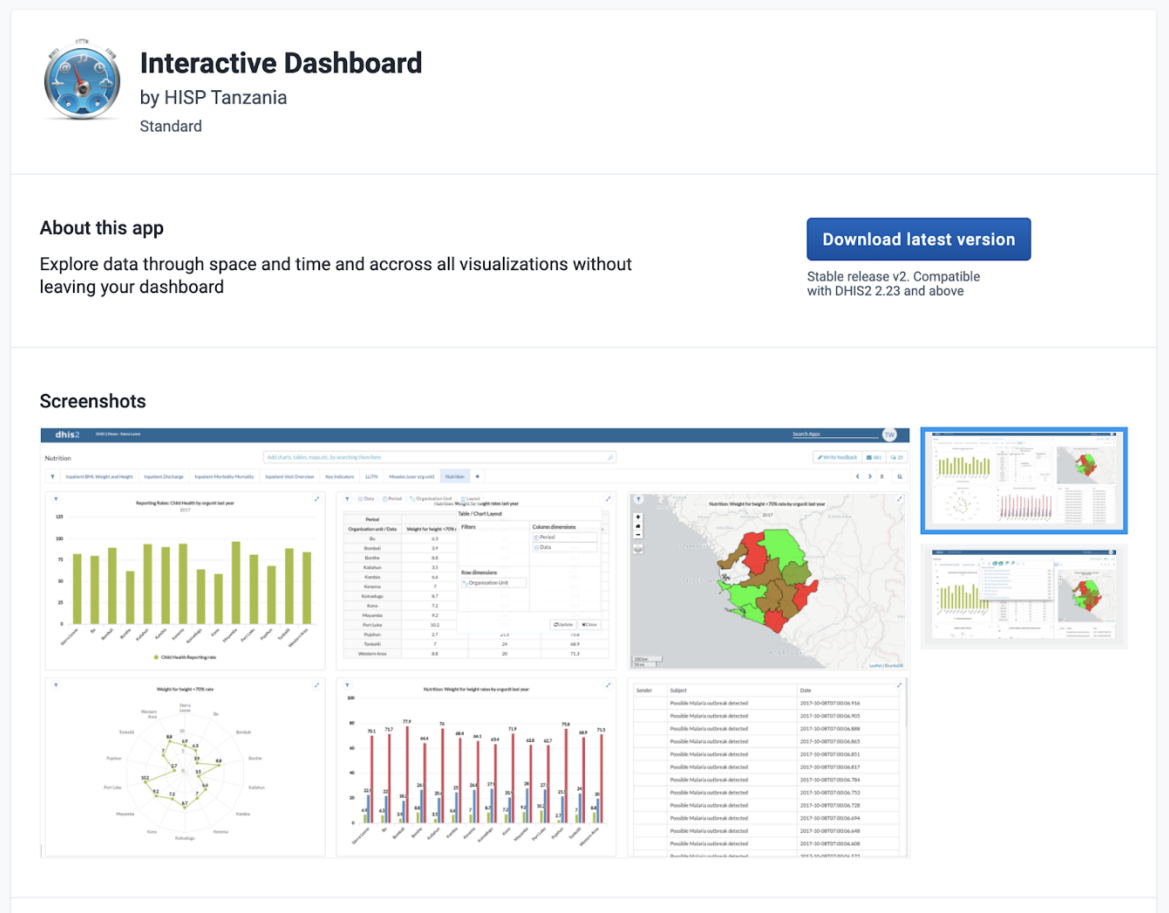


Image 4-7: HISP Tanzania's custom application

This became highly popular when appearing on the App Hub and over time the functionality was lifted into DHIS2, relieving HISP Tanzania of their application maintenance and making the same functionality more available to more user organisations through the

generic dashboard functionality. The usefulness of this application along with the needs it covers would not have been discovered as easily, had it not been available on the AppHub. While third-party app development and monitoring for useful applications on the AppHub is an effective arrangement for supporting the design of DHIS2, the vendor lacks intricate knowledge about the use of the features developed. To solve this challenge, two more arrangements were introduced, where direct user focus was the focal point.

4.4.5 Direct engagement with user organisations

In the early stages of DHIS2, when it was only implemented in a few user organisations, the core team spoke directly to all of them to develop knowledge. As this arrangement was substituted by other more efficient ways of gathering needs such as the field ambassador arrangement, these direct contact points were lost. As noted by one DHIS2 core team member, closer contact with user organisations affords important granular details used to support the design of DHIS2. In actively trying to encourage more contact, the core team has a set of arrangements focusing on direct engagement with user organisations. Here, the DHIS2 core team works together with a single organisation, gathering more granular details, such as information about clinical workflow, that feed into the process of developing knowledge to support the design of DHIS2. This form of engagement is organised in two ways: i) Retrieving feedback and needs directly from user organisation and ii) Using strategically chosen implementations for testing new functionality. While different in scope and focus, the overall goal of these approaches is to develop knowledge that supports the design of the DHIS2 by learning and gathering more granular needs directly from the context in which DHIS2 is implemented.

4.4.5.1 Specific needs from user organisations

While several arrangements in the knowledge infrastructure already ensure a general sense of the needs among user organisations, the core team lacks more intricate knowledge about how the software is used in the various implementation sites. In response to this, the core team established a direct stream of communication with a selected group of local health workers in rural Rwanda. Here, the core team is directly engaging in a WhatsApp messaging group where people from HISP Rwanda can raise their concerns and share their needs directly with a core team member. The information gathered is unfiltered and not a product of several layers of alignment processes between multiple field ambassadors. Rather, it is

based on pictures and messages of their day-to-day work, where they address some of their key insights as well as their everyday struggles. This direct way of communicating gives the core team an unfiltered view, allowing them to learn and gather insights in the process.

Rwanda being the country selected as a site for direct engagement was not accidental, but rather a strategic choice based on their relevance and qualities. As one informant described; *"First of all, HISP Rwanda is very organised. Rwanda as a country has a lot of qualities that we would like to reinforce with other nations as well"*. The Rwandan use context is a source of inspiration for well-managed DHIS2 implementations. Therefore it is valuable to use its needs for the design of DHIS2, allowing other nations to reap the benefits. The knowledge developed from Rwanda differs from the knowledge developed from Jira and COP. It gives insight to work practices on a clinical level which the digital systems do not provide. An informant described the relationship with implementations like Rwanda:

"The Rwanda use case has really helped us understand the clinical workflow. We always get a general sense of how people are moving around the different apps, but we don't get a moment-to-moment timeline of exactly what they are doing. What exactly are users doing in the system? How are they going from feature to feature? What are they inputting? What are they consuming? That's what the Rwanda project has helped shed light on. And of course, it's just a handful of clinics in one country, but it is a good point of reference."

While user organisations like the Rwandan national health system are useful points of reference for designing DHIS2, it is not equally useful for acquiring feedback on new functionality. The core team needs direct input to make sure the new generic features are fit to use by the user organisations they are designed for. This is challenging due to user organisations like the one in Rwanda having limitations in their technical infrastructure, making software upgrades and adding new features for testing rather slow. With the software mostly being implemented in low- and middle-income countries, this challenge does not only apply to implementations like Rwanda but is present in a majority of places the software is implemented. As a result, the DHIS2 core team had to introduce a new arrangement for testing new features that are being developed.

4.4.5.2 Directly testing new functionality

Acquiring a rich set of needs is important to support the initial development of a feature, fast and direct feedback is eminent to ensure its continuation and relevance. To do so, new functionality is being tested on a selected group of user organisations before being released to the whole ecosystem. An informant described why the same groups used for gathering needs are not necessarily used for feedback on the needs they provided:

"What we lack from things like Rwanda is closing the circle. We get needs from them and get them into a feature, we release them in the 2.35 release, but will they actually be useful to Rwanda? Will they actually be utilised? Those are questions we are not going to be able to answer for quite some time because Rwanda won't be able to upgrade to 2.35 for probably about a year."

As described, the core team struggles with timely feedback on new features due to many countries being several versions behind the latest release. This halts development as the core team is dependent on rapid feedback from user organisations to continue development. The process of feature development in DHIS2 can be divided into two parts: One, developing an initial product that encompasses the most general features acquired through arrangements in the knowledge infrastructure. Two, continuously develop and extend this initial product based on the feedback that they acquire from its initial use. "Closing the circle" mentioned by the informant is then referring to fully finishing a feature and not stopping development at the first stage due to time lag from countries not ready to upgrade their software. If no country is capable of upgrading, making finishes to new features could be put on hold for several years. To solve this, the DHIS2 core team seeks to be very explicit about what goes into the next release. This is to allow countries to prepare for upgrading their software. They then locate implementation sites that are capable of installing the upgrade at the time of release and use these as reference points for further development. The focus here is not to develop knowledge about every aspect of a new feature, but it is rather an arrangement to get an idea of where to go next. Based on feedback from only a handful of facilities or districts, the core team can ensure the continuation of the feature development.

4.5 Chapter summary

Figure 4-6 illustrates the whole timeline of all the arrangements from start to finish and shows what challenges the DHIS2 core team encountered along the way. In summary, the increasing number of user organisations has pushed the vendor into employing new arrangements for developing knowledge to support the design of DHIS2. The global expansion made developing knowledge more challenging as user organisations become less accessible. To overcome these, the DHIS2 core team had to change the knowledge infrastructure by either introducing new arrangements or updating existing ones. In phase one, developing knowledge was primarily based on core team members travelling to user organisations and gathering needs directly, but this was not feasible with the number of user organisations growing. In phase two, HISP groups with field representatives emerged. This proved more efficient, but over time became too disorganised with needs coming in from any user organisation. In phase three, more structure and hierarchy was introduced through the field ambassador arrangement but this distanced the DHIS2 core team too far from the user organisations. Phase four introduced more arrangements that focused on making the core team more available to user organisations as well as introducing arrangements with direct lines of communication to chosen user organisations.

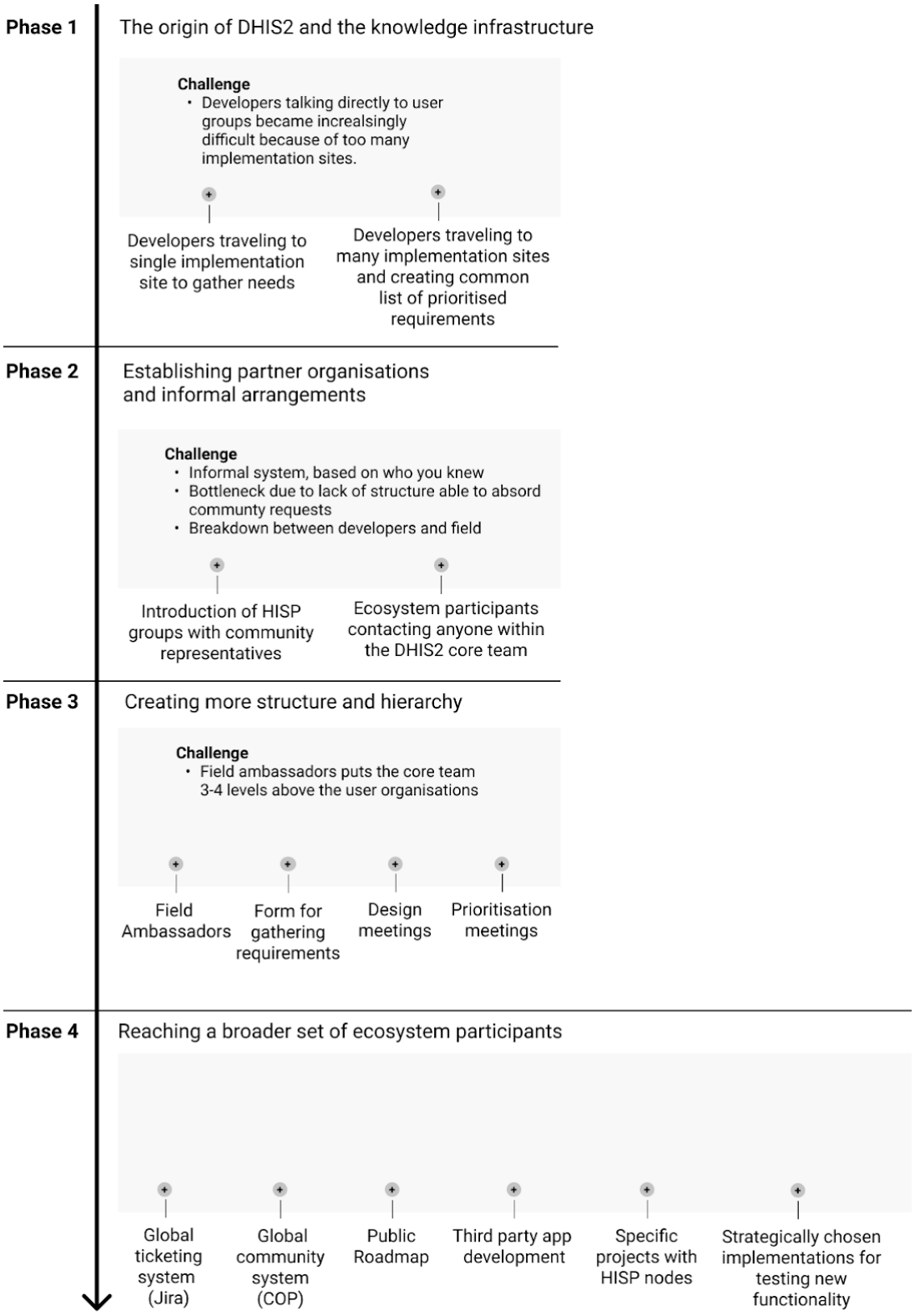


Figure 4-6: Summary of all four phases

Table 4-2 below summarises the arrangements from phases 3 and 4. These will be further used for analysis and discussions.

Arrangement	Description
Field ambassador	Representatives have an elevated status within a HISP group, where they are in charge of gathering needs and forwarding them to the DHIS2 core team while also being in charge of prioritising and filtering the gathered needs into a ranked list.
Form for gathering needs	The form which the field ambassadors are to use for gathering needs. It requires the field ambassador to fill in a description of the need, information about who it is for and what part of the generic software it involves.
Design meetings	HISP groups and user organisations are weekly invited to discuss and give feedback on features that are in development
Prioritisation meetings	Biannual meetings where all of the HISP groups are invited to participate in presenting their needs as well as have their say in what needs should be prioritised.
Jira	A global ticketing system that allows any participant within the ecosystem to request changes, report bug fixes or address the need for new functionality.
Community of practice	The community of practice (COP) is a globally available website, where user organisations engage in different topics of discussion. The website is centred around user support, capacity building and resource sharing
Public Roadmap	A timeline displaying the release plan for the upcoming version and development plans for each product stream.
Third-party app development	Find third party apps and make them generally applicable to include them as a part of the generic software.
Direct contact with Rwanda	A direct stream of communication with a selected group of local health workers where they can raise their concerns and share their needs directly to a core team member.
Directly testing new functionality	New functionality is being tested on a selected group of user organisations before being released to the whole ecosystem

Table 4-2: Summary of arrangements

5 Analysis and discussion

In this thesis we set out to address the question; *How do vendors cultivate a knowledge infrastructure for supporting the design of generic enterprise software in software ecosystems?*

Our study has focused on examining a vendor's challenges and the arrangements (i.e their knowledge infrastructure) for developing knowledge supporting the design of an ES, and how this has evolved over time. Through our analysis, we have identified four considerations for vendors when cultivating a knowledge infrastructure to support the design of ES. In this chapter, we will first address the question by articulating the four considerations for cultivating a knowledge infrastructure based on our empirical findings, while discussing their broader relevance beyond our empirical case using prior academic literature. Secondly, we will discuss the relation between these considerations and summarise our contribution.

5.1 Four considerations for cultivating a knowledge infrastructure

In the following section, we present and discuss the four considerations identified. Table 5-1 gives an overview of the four considerations for cultivating a knowledge infrastructure, a description of how they support the design of ES, and examples from our empirical findings.

Considerations for vendors	Description	Example from empirical findings
Developing a generic mindset among implementation partners and user organisations	Mitigating particularities in the flow of information from implementation partners and user organisations by making them aware that needs are only served when of generic relevance.	The public roadmap explicitly states that priority is given to generic needs that can have a broader impact, making user organisations and implementation partners aware of what to expect when voicing their specific needs. Having meetings in public forums, where the participants become more aware of the level of diversity between them, thus realising that the best way to realistically accommodate their specific needs is through aligning them with others.
Maintaining a distanced comprehension of user organisations	Developing knowledge on user organisations while maintaining a distance to avoid commitments serving their specific needs	Outsourcing alignment work to field ambassadors, affording for a cost-effective gathering of information about needs while maintaining the right distance.
Strengthening the absorptive capacity for externally produced knowledge	The vendor's ability to identify and absorb externally produced innovation outputs, as well as valuable information residing beyond the reach of the vendor.	By introducing global communication channels such as Jira and COP, the DHIS2 core team is limiting the barriers of participation and sharing of knowledge between them and the broader community of user organisations.
Establishing strategic partnerships with user organisations	Forming close relationships with a few strategically chosen user organisations, acting as points of reference for understanding specific work practices and testing usability of the generic software.	Particularities acquired from a few chosen user organisations are used as reference points for understanding clinical workflows. Direct contact with user organisations for gathering rapid feedback on features in development.

Table 5-1: Considerations for vendors when cultivating their knowledge infrastructure towards supporting the design of ES.

5.1.1 Developing a generic mindset among implementation partners and user organisations

The first consideration we identify is *developing* a *generic mindset among implementation partners and user organisations*. When cultivating a knowledge infrastructure, vendors should consider how one could mitigate particularities in the flow of information from implementation partners and user organisations by making them aware that needs are only served when of generic relevance. By doing so, a vendor saves time and resources in the process of filtering and aligning the needs to develop knowledge about their diverse set of user organisations. One of the most prominent challenges the DHIS2 core team dealt with was managing both the amount of information and the incompatibilities within the diverse needs communicated to them. Over time, the DHIS2 core team employed arrangements both focusing on filtering and aligning needs coming from the evolving set of diverse user organisations in order to develop knowledge supporting the design of DHIS2. The various meetings and field ambassador arrangement are examples of this. These arrangements are directed towards having a knowledge infrastructure supporting the DHIS2 core team in being able to comprehend the incoming information. This focus draws similarities to efforts of alignment discussed in existing literature (Pollock & Williams, 2007; Ulriksen, 2017). However, our empirical findings also show that a consideration behind these arrangements is directed towards actively gearing the various implementation partners and user organisations into thinking in more generic terms. The DHIS2 core team is trying to develop a generic mindset among the implementation partners and user organisations in the ecosystem, and several of the arrangements of the knowledge infrastructure play an important role in achieving this.

One example of this is the meetings occurring between the core team and representatives from various implementation partners and user organisations, focusing on discussing the ranking of needs or feedback on new features. These meetings are not conducted in a private one-to-one manner, but rather in more public forums where the participants engage with each other as well as the DHIS2 core team in communicating their various needs and requests for changes. This collective form of gathering and discussing needs can be seen as facilitating for an effect similar to what Pollock & Williams (2007) calls “witnessing”, where

the participants become more aware of the level of diversity between them, thus realising that the best way to realistically accommodate their specific needs is through aligning them with others. The global communication channels of JIRA and Community of Practice can also be seen as facilitating similar witnessing effects. Being globally available websites, various implementation partners and user organisations can both engage in, and witness, the level of diversity in requested changes and discussions. Thus, JIRA and COP also contribute to actively gearing the participants on these websites into thinking in more generic terms.

Our findings also show that this "generic mindset" is not only pushed in conversational settings with representatives and through the global communication channels. This is also promoted explicitly through arrangements such as the public roadmap and the form for formulating the needs of user organisations, standardising how needs are reported. The public roadmap explicitly states that priority is given to generic needs that can have broader impacts for multiple user organisations. This is similar to observations made by Campagnolo et al. (2015, p. 157), where an important strategy of managing differences between user organisations was directed towards "*preparing them to conceive of their needs in a form that is generic enough to fit the whole user community*". Clearly stating their "generic mindset" might also further help with expectation management, making user organisations understand and adapt to what they can expect when voicing needs for various generic software features.

5.1.2 Maintaining a distanced comprehension to user organisations

The second consideration we identify is *maintaining a distanced comprehension of user organisations*. When cultivating a knowledge infrastructure, vendors should consider how one could develop knowledge on user organisations while maintaining a distance to avoid commitments serving their specific needs. In line with Sawyer (2001), we argue that the close connection between the vendor and the user organisations when designing bespoke software has been replaced by an arms-length relationship when designing an ES. A persistent challenge we found in our empirical findings was the DHIS2 core team struggling to develop knowledge on a set of diverse needs without becoming too obligated to serve the needs of specific user organisations. This challenge is found in other studies such as

Campagnolo et al., who notes *"Learning how not to become too embedded in the context of a practice is therefore, one neglected social form of extended practice"* (2015, p. 158). Literature has also noted how becoming too embedded when developing knowledge to support the design puts the ES at risk of being tied to a specific user organisation or practice (Bansler & Havn, 1994) This may result in the ES becoming less generic as it complies too closely with one user organisation's way of working, and thus not be as widely marketable (Pollock, 2005). Alternatively, other user organisations may also become neglected in favour of one chosen user organisation used as the only source of information (Wagner et al., 2006).

Similar to Pollock et al. (2007) we see that a distance to user organisations is maintained by the vendor through a hierarchy of intermediaries. Pollock et al. (2007) specifically describe how developing knowledge to support the design of ES is based on representatives from user organisations functioning as "surrogates" for modelling the software before it is released as a global solution. The field ambassador arrangement we describe compares to this, as the core team is using intermediaries, acting as representatives for a larger set of user organisations. This way, the core team can acquire needs from a large set of user organisations at a distance in a cost-effective manner as portions of the vendor work is outsourced to others. While this system efficiently provides the vendor with information about needs, the vendor is no longer involved in aligning needs coming from a majority of the user organisations. This implies less control over what type of needs that ends up influencing the design of DHIS2. While the core team may deem this absence of control undesirable, the distance is necessary to ensure that a broader set of user organisations is being represented in the process of designing the ES. Maintaining a distanced comprehension of user organisations allows the core team to develop knowledge to support the design of DHIS2, while not becoming too embedded in the context of interest.

5.1.3 Strengthening the absorptive capacity for externally produced knowledge

Our third consideration we identify is *strengthening the absorptive capacity for externally produced knowledge*. When cultivating a knowledge infrastructure, vendors should consider how one could strengthen the ability to identify and absorb externally produced innovation outputs, as well as valuable information residing beyond the reach of the vendor.

A challenge facing the DHIS2 core team over the years is the continuous struggle to effectively communicate with its larger community of user organisations. While the use of intermediaries in the field ambassador system allowed for a broader set of user organisations to be represented in the process of developing knowledge to support the design of DHIS2, the number of user organisations the field ambassadors could confer with was still limited. As the number of user organisations continued to increase, the barriers of communication between the DHIS2 core team and the broader set of user organisations grew accordingly. Thus, within the growing community of user organisations, an increasing amount of information resided beyond the reach of the DHIS2 core team. In fear of missing out on potentially valuable information due to this growing communication gap, we see that a key consideration in cultivating the knowledge infrastructure is strengthening the absorptive capacity by introducing several global communication channels. Roberts et al. (2012) define the notion of absorptive capacity as the ability to identify and utilise externally produced knowledge, i.e. all knowledge generated without the vendor being included in the processes (e.g knowledge exchanged between user organisations or a third-party app made by an implementation partner). By introducing the global communication channels such as Jira and COP and the App Hub, the DHIS2 core team is limiting the barriers of participation and sharing of knowledge between them and the community of user organisations. Our findings show that a consideration behind these arrangements was strengthening the absorptive capacity in terms of both the ability to identify innovation outputs, as well as information about challenges and needs.

Firstly, our empirical findings show that due to an increasing number of user organisations with specific needs, multiple custom configurations and applications are being made. This development draws similarities to accounts made by Von Hippel (2005), highlighting that users turn to innovative solutions in response to their specific needs not being met by a vendor. With this evolution, the emerging challenge for the vendor then becomes absorbing and capitalising on the outputs coming from these innovation processes. As seen in our empirical findings, the DHIS2 core team introduces structures focused on tackling this challenge, offering an App Hub for various implementation partners and user organisations to share their innovations, occasionally reverse-engineering innovations into the ES. This is in line with extant literature, highlighting the importance of tapping into local innovations across the ecosystem (Gizaw et al., 2017; Dittrich, 2014).

Secondly, our empirical findings also show that absorptive capacity was not solely directed towards higher innovation uptake. The DHIS2 core team also introduced a global ticketing platform in Jira, and a community of practice website, facilitating for easier sharing of needs and challenges, as well as engaging in different topics of discussion. These global communication channels afford a more direct line of communication both between the user organisations and the vendor, but also between the user organisations themselves as they often indulge in helping each other on the websites. This is in line with Mozaffar (2016), highlighting the diverse roles that arrangements similar COP and Jira can take, such as being an effective up-to-date informant for the vendor, where the engagements between participants can lead to the identification of common needs or challenges.

Where the technologies facilitating the easier exchange of knowledge is an important factor for the absorptive capacity, they are not responsible for the success in itself. Prior literature has highlighted the positive impact of global communication channels on a vendor's ability to collect vast amounts of externally produced knowledge (Gao et al., 2017). Yet, reaping value from such arrangements is also a result of how they are utilised and the structures put in place for facilitating the absorption of knowledge residing beyond the reach of the vendor. One example of this shown in our empirical findings is the utilisation of Jira. While being an effective source to accumulate accessible externally produced knowledge, the amount of information makes it difficult to identify what can be directly valuable in supporting the design of an ES. Thus, the DHIS2 core rather uses Jira to complement prior knowledge accumulated from other arrangements within the knowledge infrastructure.

5.1.4 Establishing strategic partnerships with user organisations

The fourth consideration we identify is *establishing strategic partnerships with user organisations*. When cultivating a knowledge infrastructure, vendors should consider forming close relationships with a few strategically chosen user organisations, acting as points of reference for understanding specific work practices and testing the usability of the generic software features. A challenge present in both extant literature and our findings relates to acquiring contextual knowledge from a large and growing number of user organisations to support the design of an ES (Johnson et al., 2014). While we previously observed that the DHIS2 core team kept an arms-length relationship with the majority of

the user organisations, we also saw that acquiring context-specific knowledge was central in developing knowledge to support the design of DHIS2. We see that establishing strategic partnerships was a key consideration in cultivating the knowledge infrastructure. The DHIS2 had a focus on introducing arrangements providing context-specific knowledge, usable beyond the settings from which it is gathered from.

In the literature concerning the design of ES within software ecosystems, there is a broad sense that developing knowledge relates to the alignment of diverse needs, and thus mitigating particularities is a necessity (Pollock et al., 2007; Silsand & Ellingsen, 2014; Koch, 2007; Bansler & Havn, 1996). However, our empirical findings show that vendors also purposely include arrangements that ensure contextuality as a part of the knowledge infrastructure. It further shows that the particularities acquired from a few user organisations are used as reference points for understanding the workflow of various user organisations. Specifically in the case of the WhatsApp group Rwanda. In that setting, the particularities are not used for alignment nor neglected when supporting the design of DHIS2, but instead used as a direct resource for developing or improving it. Another example of this is the strategically chosen user organisations the DHIS2 core team used for gathering rapid feedback on newly developed software features. The feedback coming from this arrangement did not exclusively inform the software features. Rather, it complemented the outcomes of other arrangements with more granular details on the usability of the planned software feature. Wagner et al. (2006) criticise this way of developing ES as only a few selected points of input speak on the behalf of a global set of user organisations. While Wagner et al. (2006) is partly right in their claim, our analysis shows that these types of arrangements do not exclusively inform the design. They are a part of a larger knowledge infrastructure enriched by a broad set of arrangements substantiated by the considerations provided by this thesis.

5.2 Extending current knowledge on knowledge infrastructures

As the analysis shows, these considerations for cultivating a knowledge infrastructure can be seen as somewhat conflicting. One example is establishing close partnerships with certain user organisations (*strategic partnerships*), while not wanting to be too embedded in specific contexts of use (*distanced comprehension*). However, we argue that they are complementary, all essentially serving different roles in cultivating a knowledge infrastructure best enabled to support design of an ES.

Developing a generic mindset can be seen as necessary for maintaining a level of distanced comprehension. As both prior literature and our analysis show, developing knowledge without becoming too embedded in the context of interest, is often achieved through having arrangements focusing on some form of alignment work (Pollock & Williams, 2007; Silsand & Ellingsen, 2014; Campagnolo et al., 2015). As alignment work involves a process whereby specific needs are lifted out of their context and abstracted into 'generic' needs, mitigating the number of specific needs involved in this process is therefore important to ensure its effectiveness and success. This was especially relevant in the case of DHIS2, due to the majority of alignment work being outsourced to the field ambassadors. Prior literature has conceptualised alignment work as unfolding directly between the vendor and strategically chosen intermediaries representing a broader set of user organisations (Fruijtier & Pinard, 2017). However, we discovered that the field ambassadors also developed knowledge about the user organisations through engaging in alignment processes of their own. This could be described as a form of *outsourced generification* where the field ambassadors conduct data gathering and alignment beyond the DHIS2 core teams control. While this new way of using intermediaries affords for developing knowledge about a broader set of user organisations in a cost-effective manner, it comes at the expense of control. Thus, developing a generic mindset, making the participants involved in these processes aware that needs are only served when of generic relevance, complements the consideration of maintaining a distanced comprehension.

While developing a generic mindset, and maintaining a distanced comprehension is important for the necessary alignment work required to develop knowledge about a diverse set of user organisations, these considerations do not come without challenges. As present

in our analysis, these considerations might introduce barriers of communication between a vendor and its larger community of user organisations, as well as the loss of potentially important contextual knowledge. The consideration of strengthening the absorptive capacity and establishing strategic partnerships is therefore important in complementing the focus on developing generic mindsets and maintaining distanced comprehension.

The consideration of strengthening the absorptive capacity complements the focus on maintaining a distanced comprehension, by having a knowledge infrastructure also directed towards being able to identify and utilise valuable external information residing beyond the reach of the vendor. Arrangements focusing on strengthening the absorptive capacity should also take into account the diversity of roles a larger community of user organisations and implementation partners can have, both as a source of innovation outputs and as an effective up-to-date informant of needs and challenges (Mozaffar, 2016). Strategic partnerships complement the focus on developing generic mindsets among user organisations with more contextual knowledge. Arrangements focused on having reference points for understanding specific work practices gives insight into the usability of an ES, contributing to informing the granular details of generic software features. Similar one-to-one relationships between the vendor and a specific user organisation have been conceptualised in prior literature. Johnson et al. (2014) describe how strategic partnerships are directly informing the design of the ES. However, these strategic partnerships are highlighted as only being used in the initial stages of vendors entering new markets, in situations where prior knowledge of existing user organisations was not transferable to the new context (Johnson et al., 2014, p. 806). Beyond what is discussed in the literature, our analysis shows that strategic partnerships are used to continuously complement the more aggregate and "generic" understanding coming from other arrangements with more contextual knowledge. One example is the core team selecting a few user organisations to test the usability of a software feature. The rapid feedback provided the core team with details on how it affects their workflow. Contextual knowledge is oftentimes seen as hinders to developing knowledge about a diverse set of user organisations (Koch, 2007). Our case provides insight into ways in which strategic partnerships are continuously used for supporting the design of an ES.

5.3 Contribution

The contribution of this thesis is both practical and theoretical. We relate the theoretical contribution to the literature concerning the design of generic enterprise software within ecosystems. In our chapter on related literature, we noted that prior literature has highlighted a series of aspects related to developing knowledge about a diverse set of user organisations to support the design of ES. Conceptualising the collective of arrangements used by the vendor to develop knowledge about a diverse set of user organisations, Johnson et al. (2014) introduced the notion of knowledge infrastructures, highlighting how industry practices were far more intricate than extant literature would suggest. Concurring with Johnson et al. (2014), our study has focused on examining a vendor's challenges in developing knowledge, the collective of arrangements they introduced to address these (i.e their knowledge infrastructure), and how this has evolved. From this, we make both a practical and theoretical contributions. Firstly, we offer a practical contribution to the DHIS2 core team, addressing the need for an overview of the various arrangements for developing knowledge discovered through an iterative process of problem formulation from our engaged research project. By mapping out their current arrangements for developing knowledge, we provide them with insight on how each arrangement contributes to developing knowledge about their diverse set of user organisations.

Secondly, we make a theoretical contribution by extending the notion of knowledge infrastructure using concepts from IS infrastructure theory. Central to this theory is the argument that the success of an information infrastructure must be achieved over time by cultivating growth and adapting to complexity (Hanseth & Lyytinen, 2010). In similarity, a knowledge infrastructure is not designed by selecting and putting together arrangements based on fixed goals with planned outputs (Bergqvist & Dahlberg, 1999; Hanseth & Lyytinen, 2010). ES vendors must rather attempt to incrementally adapt their knowledge infrastructure according to the changes happening within the ecosystem. As present in both our empirical findings and literature, the complex dynamics of a software ecosystem continuously challenge a vendor's efforts in keeping the ES as useful and relevant as possible for a diverse set of user organisations (Dittrich, 2014; Johnson et al, 2014). As the contexts of use are not fixed, but evolve, we conceptualise knowledge infrastructure as subject to ongoing cultivation by the vendor in order to improve their ability to develop knowledge

about the diverse set of user organisations. Extending the notion of knowledge infrastructure using concepts from IS infrastructure theory has further contributed to exploring how a vendor works to develop knowledge about their diverse set of user organisations to support the design of an ES. Viewing the collective of arrangements used by a vendor to develop knowledge as a knowledge infrastructure subject to ongoing cultivation, opened up the scope of analysis on our empirical data, contributing to underpin the four considerations for cultivating a knowledge infrastructure. We argue that our extension of knowledge infrastructures provides a conceptualisation for capturing other vendors' efforts in developing knowledge about their diverse set of user organisations to support the design of an ES.

Our analysis further offers a theoretical contribution by identifying four considerations for a vendor when cultivating a knowledge infrastructure to support the design of ES. While extant literature has addressed some aspects related to developing knowledge about a diverse set of user organisations to support the design of ES (Pollock et al., 2007; Koch, 2007; Campagnolo et al., 2015; Joshi et al., 2007; Gizaw et al., 2017), the four considerations for a vendor in cultivating a knowledge infrastructure represents a novel contribution to this body of knowledge. For one, literature has shown how supporting design of ES within software ecosystems requires a focus on the general needs (Sia & Soh, 2007; Strong & Volkoff, 2010; Pollock et al 2007; Silsand & Ellingsen, 2014), as well as avoiding becoming too obligated to serve the needs of specific user organisations (Koch, 2007; Campagnolo et al., 2015). We contribute to these observations, highlighting how vendors face these challenges cultivating towards (1) arrangements allowing them to maintain a distanced comprehension to user organisations and (2) develop a generic mindset, making the implementation partners and user organisations aware that needs are only served when of generic relevance. Additionally, literature has also highlighted the importance of taking innovation outputs and information residing beyond the reach of the vendor into account (Dittrich, 2014; Gizaw et al., 2017; Mozaffar, 2016), as well as argued for a more persistent focus on local work practices and the diversity of user organisations (Joshi et al., 2007; Oudshoorn el al., 2004). We extend existing observations by highlighting how vendors face these challenges by cultivating towards (3) strengthening absorptive capacity for externally produced knowledge, and (4) establishing strategic partnerships with user organisations, continuously complementing other arrangements with knowledge

on specific work practices and usability of the generic software features. Thus, the four considerations extend the current understanding of the phenomenon of knowledge infrastructure by providing insight into key considerations a vendor takes in cultivating their knowledge infrastructure, with the aim of supporting the design of ES. Additionally, the four considerations are identified through a combined effort of analysing both our empirical findings, as well as discussions with cases from extant literature. From the literature, we note similarities in the challenges vendors experience in developing knowledge to support the design of ES within software ecosystems (e.g Dittrich, 2014; Johnson et al., 2014; Pollock et al., 2007). Thus, we argue that the four identified considerations are relevant for vendors beyond the DHIS2 core team, offering a practical contribution to vendors grappling with cultivating their knowledge infrastructure towards better supporting the design of their ES.

Table 5-2 summarises our contribution concerning the challenges experienced by the DHIS2 core team in developing knowledge to support the design of DHIS2, as well as challenges highlighted in related literature.

Practical and theoretical challeng		Consideration for cultivating a knowledge infrastructure	How it affects developing knowledge to support the design of ES
Challenge from existing literature	Developing software meant to fit across an array of use contexts requires a focus on the general needs (Sia & Soh, 2007; Strong & Volkoff, 2010; Pollock et al 2007).	**Consideration number 1:** Developing a generic mindset among implementation partners and user organisations	Mitigating particularities in the flow of information from implementation partners and user organisations by making them aware that needs are only served when of generic relevance.
DHIS2 challenge	DHIS2 core team struggles with managing both the amount of information and the incompatibilities within the diverse needs communicated to them.		
Challenge from existing literature	Becoming too embedded in a single practice puts the generic software at risk of being tied to a specific organisation or practice (Pollock, 2005; Koch, 2007).	**Consideration number 2:** Maintaining a distanced comprehension of user organisations	Developing knowledge on user organisations while maintaining a distance to avoid commitments serving their specific needs
DHIS2 challenge	DHIS2 core team struggling to develop knowledge on a set of diverse needs without becoming too obligated to serve the needs of specific user organisations.		
Challenge from existing literature	There is a considerable degree of innovation and knowledge within the ecosystem residing beyond the reach of the vendor (Dittrich, 2014; Gizaw et al., 2017).	**Consideration number 3:** Strengthening the absorptive capacity for externally produced knowledge	The vendor's ability to identify and absorb externally produced innovation outputs, as well as valuable information residing beyond the reach of the vendor.
DHIS2 challenge	The DHIS2 core team struggles to effectively communicate with its larger community of user organisations.		
Challenge from existing literature	“Design for everybody” might lead to “design for nobody”. A lack of focus on local work practices and the diversity of needs affords challenges in the process of implementing the ES (Oudshoorn et al., 2004; Joshi et al., 2007).	Consideration number 4: Establishing strategic partnerships with user organisations	Forming close relationships with a few strategically chosen user organisations, acting as points of reference for understanding specific work practices and testing usability of the generic software.
DHIS2 challenge	The DHIS2 core team struggles to acquire knowledge directly from a large and growing number of user organisations as the scale of the ecosystem increases.		

Table 5-2: Considerations in relation to challenges from literature and empirical findings

5.4 Limitations and further research

In line with Johnson et al., (2014), Koch (2007) and Pollock et al. (2007), the research of this thesis adds to the argument that industry practices for developing knowledge about a diverse set of user organisations are far more intricate than early conceptualisations would suggest. Our study has focused on examining the DHIS2 core team's challenges, as well as their arrangements (i.e their knowledge infrastructure) for developing knowledge to support the design of ES, and how this has evolved. We have extended the notion of knowledge infrastructure using concepts from IS infrastructure theory and identified and discussed four considerations for vendors in cultivating a knowledge infrastructure to support the design of ES. In this section, we describe the limitations that can affect our contribution and provide avenues for further research.

5.4.1 Limitations

As we have attempted to understand this social phenomenon through the intersubjective meanings and experiences of our informants, our subjectivity has naturally affected both the process of gathering and analysing our empirical material. We must also acknowledge that our engaged role with the DHIS2 core team, generating data and findings based on interpretations, makes the study hard to replicate in detail. Yet, in line with the tradition of interpretive research, this is not the main objective. As noted by Flyvbjerg, the view that one cannot generalise and make a valuable contribution based on a single case study is common. However, critiquing this view as narrow, he writes; "*That knowledge cannot be formally generalized does not mean that it cannot enter into the collective process of knowledge accumulation in a given field or in a society.*" (Flyvbjerg, 2006, p. 227). We argue for the relevance of our considerations beyond the case of DHIS2 as they are identified through a combined effort of analysing our empirical findings, as well as discussions with cases from extant literature. To establish credibility of these results, we have strived to provide a rich description of the methods for data collection and analysis, as well as our findings to enable other researchers to follow the arguments that have led us to our contribution.

Other limitations may be found in how we conducted our research project. We utilised various forms of data gathering activities, where the primary source of information has been interviews and meetings with members of the DHIS2 core team. We have enriched our understanding and mitigated the likelihood of misinterpretation by combining interviews with observations and documents analysis. Yet, our study could benefit from having conducted more observations of the interactions between the DHIS2 core team, HISP groups and user organisations. One example is the design meetings and prioritisation meetings where representatives from HISP groups and user organisations are invited to voice their needs for new software features or give feedback on features that are in development. Conducting more observations of the interactions taking place in these meetings could provide us with a richer understanding of what goes into developing knowledge to support the design of an ES.

5.4.2 Avenues for further research

Based on the findings of our study we suggest several avenues for further research. First, our study is limited to examining the knowledge infrastructure of one vendor within a software ecosystem. Conducting a single-case study, our empirical findings are naturally shaped by the DHIS2 core team's organisational characteristics. We have shown how one vendor has grappled with cultivating a knowledge infrastructure, growing from a one-country implementation to an ES used in more than 70 countries globally, and identified four considerations taken into account in this process. One avenue for further research would be to investigate the relevance and applicability of these considerations beyond the case of DHIS2.

Second, highlighting the complexity of developing knowledge about a diverse set of user organisations to support the design of ES, the four considerations provide a basis for several relevant avenues for further research:

(i) Our study indicates that a generic mindset might help to manage the expectations of user organisations when voicing their needs. However, how this materialises among user organisations has not yet been explored in the detail. Of particular interest would be to further explore how the witnessing effects facilitated by global communication channels compare to physical meetings in affecting how the user organisations communicate their needs.

(ii) Our study illustrates that the vendor of DHIS2 is able to maintain a distanced comprehension to the user organisations by outsourcing portions of the alignment work to implementation partners operating closer to the user organisations. A relevant avenue for further research would be to investigate the implementation partners role in this process, specifically in terms of competence and actions taken to develop knowledge about a diverse set of user organisations.

(iii) Our analysis highlights that one way of developing knowledge is by absorbing applications from third-party developers. Recently more efforts have been put into developing DHIS2 as a platform with a greater focus on developing apps to address specific features. Our study is focused on how DHIS2 core team develops knowledge to support the design of the DHIS2 software, thus, the platform capabilities of DHIS2 has received less attention. However, as the consideration of strengthening the absorptive capacity is directed towards externally produced knowledge, one interesting avenue for further research would be to look at how one could adapt the absorptive capacity towards a platform-oriented strategy (Wareham et al., 2013).

(iv) Further, our study shows that the DHIS2 has established close relationships with a few strategically chosen user organisations, acting as points of reference for understanding specific work practices and testing usability of the generic software. One avenue for further investigation here is to understand other qualifications or attributes a vendor seeks when identifying and establishing these strategic partnerships.

Lastly, we do not claim our list of four considerations to be exhaustive. Further research investigating vendors of other software ecosystems could thus be useful for elaborating, modifying or extending the list of considerations for cultivating a knowledge infrastructure presented in this thesis. Other ES vendors might have encountered different challenges and employed other arrangements that would be valuable to uncover and add to the pool of knowledge on considerations for cultivating a knowledge infrastructure to support the design of ES. In general, we welcome anyone within the research context of ES within software ecosystems to challenge our contribution or further enhance it by either substantiating what we have discovered with more empirical findings or providing new insight that can result in new considerations.

6 Conclusion

Through examining a vendor's challenges in developing knowledge about their diverse set of user organisations, the collection of arrangements put in place to remedy these challenges and how this has evolved over time, this thesis provides insight into how a vendor cultivates a knowledge infrastructure for supporting the design of an ES. Insight into the process of how vendors cultivates a knowledge infrastructure is relevant as vendors increasingly find themselves designing for a large and heterogeneous audience of user organisations, a feat that is reported as challenging in IS literature (Strong & Volkoff, 2010; Berente et al., 2016; Sia & Soh, 2007).

Prior literature highlights various aspects related to developing knowledge about a diverse set of user organisations to support the design of ES. Due to the heterogeneity of user organisations a vendor seeks to accommodate, the vendor needs to find means of bridging a diverse set of needs (Pollock & Williams, 2007). As ES vendors cannot cater to the particularities of each user organisation, specific needs are labeled as barriers to the design of an ES (Koch, 2007). Thus, the vendor must balance between keeping a distance to avoid commitments to single user organisations (Campagnalo et al., 2015; Sawyer, 2001) and necessary interaction in order to develop knowledge reflecting the user organisations actual needs (Joshi et al., 2007). A commonality in the literature highlighting how vendors retain a distance to the user organisations is that the main points of interaction are happening between the vendor and intermediaries, communicating the needs on behalf of the larger community of user organisations (Pollock & Williams, 2007; Johnson et al., 2014; Fruijtier & Pinard, 2017). However, literature also highlight the importance of taking innovation outputs and information residing beyond the reach of the vendor into account (Dittrich, 2014; Gizaw et al., 2017; Mozaffar, 2016). Albeit its relevance to further understand how vendors manage these aspects when developing knowledge to support design of an ES, knowledge infrastructures remains an understudied phenomenon.

Adding to this discussion, we extend the notion of knowledge infrastructure using concepts from IS infrastructure theory. We conceptualise the collection of arrangements as knowledge infrastructures, subject to ongoing cultivation by the vendor to improve their ability to develop knowledge about their diverse set of user organisations to support design of an ES. With this understanding as a basis, we have analysed the empirical findings from a one year engaged research project, where we have collaborated with the vendor of a global health information system, focusing on how they have cultivated their knowledge infrastructure over time. Based on a combined effort of analysing our empirical findings, and a discussion with related academic literature, we identify four considerations for vendors when cultivating a knowledge infrastructure to support the design of an ES. We see that vendors should consider: (1) Developing a generic mindsets among implementation partners and user organisations, (2) Strengthening the absorptive capacity for externally produced knowledge, (3) Maintaining a distanced comprehension to user organisations and (4) Establishing strategic partnerships with user organisations, when cultivating their knowledge infrastructure to support the design of ES.

In line with Johnson et al., (2014), Koch (2007) and Pollock et al. (2007), our thesis adds to the argument that industry practices for developing knowledge about a diverse set of user organisations are far more intricate than earlier conceptualisations would suggest. The extended conceptualisation and four considerations provide insight into how vendors cultivate a knowledge infrastructure to support the design of an ES. This thesis offers two contributions to literature concerning design of ES within software ecosystems. First, our extended conceptualisation of knowledge infrastructures takes into account the complex dynamics of a software ecosystem, providing a conceptualisation for capturing vendors' efforts in developing knowledge about their diverse set of user organisations to support the design of an ES. Secondly, the four considerations extend the current understanding of the phenomenon of knowledge infrastructures by providing insight into key considerations a vendor takes in cultivating their knowledge infrastructure, with the aim of supporting the design of ES. In addition to being relevant for researchers, our conceptualisations and empirical findings are relevant to practitioners grappling with developing knowledge about their diverse set of user organisations to support the design of their ES.

References

Bansler, J. & Erling, H. (1996) 'Industrialised Information Systems Development', CTI Working Paper (22.), Technical University of Denmark.

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

Berente, N., & Yoo, Y. (2012). Institutional Contradictions and Loose Coupling: Postimplementation of NASA's Enterprise Information System. *Information Systems Research*, *23*(2), 376–396. https://doi.org/10.1287/isre.1110.0373

Bergqvist, J & Dahlberg, P. (1999). Scalability through Cultivation. *Scandinavian Journal of Information Systems*, Vol. 11(1) , Article 1.

Braa, J. & Sahay, S. (2017). The DHIS2 Open Source Software Platform: Evolution Over Time and Space. In L. Anthony, G. Celi, S.Hamish F. Fraser, V. Nikore, J. Osorio, K.Paik (eds.), Global Health Informatics. Principles of eHealth and mHealth to Improve Quality of Care. The MIT Press

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

Campagnolo, G. M., Pollock, N., & Williams, R. (2015). Technology as we do not know it: The extended practice of global software development. *Information and Organization*, *25*(3), 150–159. https://doi.org/10.1016/j.infoandorg.2015.06.001

Crang, M., & Cook, I. (2007). *Doing Ethnographies*. SAGE Publications Ltd. https://doi.org/10.4135/9781849208949

Davenport, H. (1998). Putting the Enterprise into the Enterprise System. Harvard Business Review, 1998(July), 121-131.

DHIS2. (n.d). About DHIS2. Retrieved 14 October 2021, from https://dhis2.org/about/

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

Flyvbjerg, B. (2006). Five Misunderstandings About Case-Study Research. *Qualitative Inquiry*, *12*(2), 219–245. https://doi.org/10.1177/1077800405284363

Foerderer, J., Kude, T., Schuetz, S. W., & Heinzl, A. (2019). Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance. *Information Systems Journal*, *29*(1), 119–144. https://doi.org/10.1111/isj.12186

Fruijtier, E., & Pinard, M. (2017). Collaborative Development of Global Information Systems: Toward Community Based Generification. *The Electronic Journal of Information Systems in Developing Countries*, *78*(1), 1–20. https://doi.org/10.1002/j.1681-4835.2017.tb00571.x

Gao, S., Yeoh, W., Wong, S. F., & Scheepers, R. (2017). A literature analysis of the use of Absorptive Capacity construct in IS research. *International Journal of Information Management*, *37*(2), 36–42. https://doi.org/10.1016/j.ijinfomgt.2016.11.001

Gizaw, A. A., Bygstad, B., & Nielsen, P. (2017). Open generification: Open generification. *Information Systems Journal*, *27*(5), 619–642. https://doi.org/10.1111/isj.12112

Grisot, M., Hanseth, O. & Thorseng, A.. (2014). Innovation Of, In, On Infrastructures: Articulating the Role of Architecture in Information Infrastructure Evolution. *Journal of the Association for Information Systems*, *15*(4), 197–219. https://doi.org/10.17705/1jais.00357

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

Hanseth, O., & Lyytinen, K. (2010). Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet. *Journal of Information Technology*, *25*(1), 1–19. https://doi.org/10.1057/jit.2009.19

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

Johannessen, L. K., & Ellingsen, G. (2009). Integration and Generification—Agile Software Development in the Healthcare Market. *Computer Supported Cooperative Work (CSCW)*, *18*(5–6), 607–634. https://doi.org/10.1007/s10606-009-9097-8

Johnson, M., Mozaffar, H., Campagnolo, G. M., Hyysalo, S., Pollock, N., & Williams, R. (2014). The managed prosumer: Evolving knowledge strategies in the design of information infrastructures. *Information, Communication & Society*, *17*(7), 795–813. https://doi.org/10.1080/1369118X.2013.830635

Joshi, S., Barrett, M., Walsham, G., & Cappleman, S. (2007). Balancing Local Knowledge Within Global Organisations Through Computer-Based Systems: An Activity Theory Approach. *Journal of Global Information Management*, *15*(3), 1–19. https://doi.org/10.4018/jgim.2007070101

Kallinikos, J. (2004). Deconstructing information packages: Organizational and behavioural implications of ERP systems. *Information Technology & People*, *17*(1), 8–30. https://doi.org/10.1108/09593840410522152

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

Koch, C. (2007). ERP a moving target. *International Journal of Business Information Systems*, *2*(4), 426. https://doi.org/10.1504/IJBIS.2007.012544

Li, M. (2019). An Approach to Addressing the Usability and Local Relevance of Generic Enterprise Software. *Selected Papers of the IRIS*, 10(3).

Li, M. (2021). Entering and Organizing Engaged Information Systems Research. Information Systems Working Paper Series at University of Oslo. Edited by Petter Nielsen. 6/2021. Retrieved from the website: http://www.mn.uio.no/ifi/english/research/groups/is/publications/working-papers-in-information-systems

Li, M., & Nielsen. (2019). *Making Usable Generic Software. A Matter of Global or Local Design?* https://doi.org/10.13140/RG.2.2.31514.49603

Light, B. (2005). Going beyond 'misfit' as a reason for ERP package customisation. *Computers in Industry*, *56*(6), 606–619. https://doi.org/10.1016/j.compind.2005.02.008

Magnusson, J., & Nilsson, A. (2013). Introducing app stores into a packaged software ecosystem: A negotiated order perspective. *International Journal of Business Information Systems*, *14*(2), 223. https://doi.org/10.1504/IJBIS.2013.056140

Mahoney, J. T. (2008). Engaged Scholarship: A Guide for Organizational and Social Research, by Andrew H. Van de Ven. Oxford: Oxford University Press, 2007. *Academy of Management Review*, *33*(4), 1015–1019. https://doi.org/10.5465/amr.2008.34424997

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

Mathiassen, L. and Nielsen, P A. (2008). Engaged Scholarship in IS Research. *Scandinavian Journal of Information Systems,* Vol. 20(2) , Article 1.

Monteiro, E., Pollock, N., Hanseth, O., & Williams, R. (2013). From Artefacts to Infrastructures. *Computer Supported Cooperative Work (CSCW)*, *22*(4–6), 575–607. https://doi.org/10.1007/s10606-012-9167-1

Mozaffar, H. (2016). User Communities as Multi-Functional Spaces. In Hyysalo, S., Jensen, T.E., & Oudshoorn, N. (Eds.). *The New Production of Users: Changing Innovation Collectives and Involvement Strategies* (1st ed.). Routledge. https://doi.org/10.4324/9781315648088

Myers, M. D. "Qualitative Research in Information Systems," MIS Quarterly (21:2), June 1997, pp. 241-242. MISQ Discovery, archival version, June 1997, http://www.misq.org/supplements/. Association for Information Systems (AISWorld) Section on Qualitative Research in Information Systems, updated version, last modified: May 12, 2021 www.qual.auckland.ac.nz

Oudshoorn, N., Rommes, E., & Stienstra, M. (2004). Configuring the User as Everybody: Gender and Design Cultures in Information and Communication Technologies. *Science, Technology, & Human Values*, *29*(1), 30–63. https://doi.org/10.1177/0162243903259190

Pollock, N. (2005). When Is a Work-Around? Conflict and Negotiation in Computer Systems Development. *Science, Technology, & Human Values*, *30*(4), 496–514. https://doi.org/10.1177/0162243905276501

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

Rickmann, T., Wenzel, S., & Fischbach, K. (2014). Software Ecosystem Orchestration: The Perspective of Complementors. *AMCIS*.

Roberts, Galluch, Dinger, & Grover. (2012). Absorptive Capacity and Information Systems Research: Review, Synthesis, and Directions for Future Research. *MIS Quarterly*, *36*(2), 625. https://doi.org/10.2307/41703470

Roland, L K., Sanner, T A., Sæbø, J I., and Monteiro, E. (2017). P for Platform. Architectures of large-scale participatory design. *Scandinavian Journal of Information Systems*, Vol. 29(2) , Article 1.

Sawyer, S. (2001). A market-based perspective on information systems development. *Communications of the ACM*, *44*(11), 97–102. https://doi.org/10.1145/384150.384168

Schneider, S., Wollersheim, J., Krcmar, H., & Sunyaev, A. (2018). How do Requirements Evolve over Time? A Case Study Investigating the Role of Context and Experiences in the Evolution of Enterprise Software Requirements. *Journal of Information Technology*, *33*(2), 151–170. https://doi.org/10.1057/s41265-016-0001-y

Seddon, Calvert, & Yang. (2010). A Multi-Project Model of Key Factors Affecting Organizational Benefits from Enterprise Systems. *MIS Quarterly*, *34*(2), 305. https://doi.org/10.2307/20721429

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

Silsand, L. & Ellingsen, G. (2014). Generification by Translation: Designing Generic Systems in Context of the Local. *Journal of the Association for Information Systems*, *15*(4), 177–196. https://doi.org/10.17705/1jais.00358

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

Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., Mcdermid, J., & Paige, R. (2012). Large-scale complex IT systems. *Communications of the ACM*, *55*(7), 71–77. https://doi.org/10.1145/2209249.2209268

Stake, R. E. (2005): Qualitative Case Studies, In Denzin, N. & Lincoln, Y. (eds.) (2005), *The Sage Handbook of Qualitative Research*. Sage Publications. Chap. 17, pp. 443-466.

Stewart, J. K., & Williams, R. (2005). The wrong trousers? Beyond the design fallacy: Social learning and the user. In D. Howcroft & E. M. Trauth (Eds.), *Handbook of critical information systems research: Theory and application* (pp. 195–221). Edward Elgar Pub.

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

Suchman, L. (2002). Located accountabilities in technology production. *Scandinavian Journal of Information Systems*, Vol. 14(2) , Article 7.
Available at: https://aisel.aisnet.org/sjis/vol14/iss2/7

Sykes, T. A. & Venkatesh, V.(2017). Explaining Post-Implementation Employee System Use and Job Performance: Impacts of the Content and Source of Social Network Ties. *MIS Quarterly*, *41*(3), 917–936. https://doi.org/10.25300/MISQ/2017/41.3.11

Tan, B., Pan, S. L., Chen, W., & Huang, L. (2020). Organizational Sensemaking in ERP Implementation: The Influence of Sensemaking Structure. *MIS Quarterly*, *44*(4), 1773–1809. https://doi.org/10.25300/MISQ/2020/11872

Titlestad, O. H., Staring, K., & Braa, J. (2009). Distributed development to enable user participation: Multilevel design in the HISP network. *Scandinavian Journal of information systems*, 21(1), 3.

UiO. (N.d-a). Health Information Systems Programme (HISP). Retrieved 10 September 2021, from https://www.mn.uio.no/ifi/english/research/networks/hisp/

UiO. (N.d-b). DHIS2 Design Lab. Retrieved 11 November 2021, from https://www.mn.uio.no/ifi/english/research/networks/hisp/dhis2-design-lab/#about

Ulriksen, G.-H. (2017). How to Involve the Users in the Large Scale Work with OpenEHR Archetypes in Norway? In Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, & S. Costanzo (Eds.), *Recent Advances in Information Systems and Technologies* (Vol. 569, pp. 757–767). Springer International Publishing. https://doi.org/10.1007/978-3-319-56535-4_74

Van de Ven, A. H. (2007). Engaged Scholarship: A Guide for Organizational and Social Research. Oxford, UK: Oxford University Press.

von Hippel, E. (2005). Democratizing innovation: The evolving phenomenon of user innovation. *Journal F□r Betriebswirtschaft*, *55*(1), 63–78. https://doi.org/10.1007/s11301-004-0002-8

Wagner, E. L., Scott, S. V., & Galliers, R. D. (2006). The creation of 'best practice' software: Myth, reality and ethics. *Information and Organization*, *16*(3), 251–275. https://doi.org/10.1016/j.infoandorg.2006.04.001

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

Wareham, J. D., Fox, P. B., & Cano Giner, J. L. (2013). Technology Ecosystem Governance. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2201688

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

Appendix 1

Informing Generic Design: Approaches to understanding a diverse user base

Conference Paper · August 2021

CITATIONS
0

READS
20

3 authors:

Sigvart Bretteville-Jensen
University of Oslo
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Jacob Ommundsen
University of Oslo
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Magnus Li
University of Oslo
14 PUBLICATIONS **30** CITATIONS
SEE PROFILE

INFORMING GENERIC DESIGN: APPROACHES TO UNDERSTANDING A DIVERSE USER BASE

Research paper

Bretteville-Jensen, Sigvart, University of Oslo, Oslo, Norway,
Sigvartb@ifi.uio.no
Kleivdal, Ommundsen, Jacob, University of Oslo, Oslo, Norway,
Jacobo@ifi.uio.no
Li, Magnus, University of Oslo, Oslo, Norway,
Magl@ifi.uio.no

Abstract

An increasingly relevant context of design of Information Technology (IT) is that of building generic enterprise software solutions, yet it remains little explored by IS research. The context is of interest as designers must deal with a large audience of user organisations with different practices and needs. Building software that sufficiently satisfies everyone is reported as challenging in IS literature. Prior studies suggest that aligning the needs of the large audience of user organisations is a key strategy employed by vendors. Adding to this discussion, we report from an action research project where we collaborate with the vendor of a globally used generic health software to document and strengthen their design practices. Based on our findings we identify and discuss three approaches a vendor leverage to inform the design of their generic solution. These are: i) Aligning needs amongst actors ii) Direct engagement with user organisations iii) Monitoring of community activity. Our study shows that alignment, although central in the vendors efforts, is but one of several elements that goes into their design efforts. We argue that our empirical account and the three approaches we discuss contribute to research and practice of the design of generic enterprise software.

Keywords: Generic Enterprise Software, Generification, Action Research

1 Introduction

Software implemented in organisations today are often 'generic' types of software, meant to fit and work across an array of organizational contexts. Common examples are Enterprise Resource planning systems (ERP´s), and Electronic Health Records (EHR). As organizational needs may vary across countries, domains and cultural contexts, a persistent challenge documented within the information systems (IS) research is developing generic systems able to meet a heterogeneous set of needs. This means that designers of IT systems increasingly find themselves in the context of designing for a large and heterogeneous audience of user organisations. Albeit its relevance, this context of design is explored little by existing IS research. Rather, focus is mainly on the challenges organisations face when implementing such solutions to support their specific practices and needs (Soh, Sia, & Tay-Yap, 2000; Pollock & Cornford, 2003).

To design and develop software, developers must find means of understanding the users practices and needs. This understanding is further used to inform design. There is a substantial body of descriptive

and prescriptive knowledge on software design and development, where various methods such as user centered design (UCD) and participatory design (PD) are commonly used (Wilkinson & De Angeli,

2014; Gregory, 2003). However, the focus is mainly on the context of 'custom' or 'bespoke' software development (Sommerville et al., 2012). These methods emphasize working closely with the user organization to understand their needs in detail. In line with others, we argue that the nature of generic enterprise software (GES) projects represents a notably different environment for design than in-house and bespoke development projects (Dittrich 2014; Sommerville et al.m 2012; Mousavidin et al. 2017). As a GES vendor considers not one, but many user organisations, achieving an understanding for design and development is correspondingly more challenging. With a large and divided user base the GES vendors often try to include features needed by many, while neglecting features only relevant to one or few. Accordingly, a GES vendor needs to gain a coherent understanding to ensure that the software complies with the large user group. To gain a coherent understanding vendors must both get an overview of the variety of needs among user organisations, while also gaining sufficiently detailed insights into the work practices they aim to support. Further, the differences and potentially incompatible needs among users must be aligned. Existing research conceptualizes important elements of how vendors inform design. Yet, a full picture of how vendors work to gain a coherent understanding to inform design of their generic solutions is lacking.

To address this gap, this paper examines the process of informing the design of a generic enterprise software solution. Concretely, the research question addressed is; *How can a vendor gain a coherent understanding of a diverse user base for informing generic design?* We address the question by examining the design and development practices of the vendor of a generic health information software named 'DHIS2'. Over time, the software has been taken into several new contexts from its initial starting point. As a result, it is increasingly challenging for the vendor to grasp and satisfy the needs of an continuously growing set of users. Thus, the vendor struggles with finding means of building a coherent understanding for informing their design. Through an analysis of practices in a global health information system the paper offers two contributions. One, by identifying a set of approaches a generic enterprise software vendor has to inform design. Two, by discussing what role different forms of approaches play in building a coherent understanding of a diverse user base for informing design.

The paper is organized as follows: Section 2 summarizes the current literature on approaches to gaining a coherent understanding and provides insight on challenges of generic design. Section 3 gives a case description, as well as describing how we conducted our research and analysis. Section 4 presents our findings. Section 5 discusses how three approaches contribute to gaining a coherent understanding for informing design. Finally, section 6 presents our conclusion.

2 Related literature

In the following section we discuss how the context of designing a GES deviates from traditional bespoke projects. We further elaborate on the challenges associated with designing generic software in regards to designing for a large and diverse audience. Finally we highlight how there is an extensive amount of literature addressing the misfits occurring between generic systems and user organisations while few have focused on what informs the generic design.

Traditional approaches for design and development of software have focused on gaining a granular understanding of user needs. A design success is often attributed to how well it fits with existing practices, routines and mental models of end-users (Norman, 2013; Strong & Volkoff, 2010). Gaining a coherent understanding of user needs is therefore crucial. In achieving this, a user-oriented focus and end-user involvement are well-established means in the process of design (Baxter & Sommerville, 2011; Rosson & Carroll, 2009). In contrast to bespoke software development, GES represents a very different context for design and development, characterized by significant variety and heterogeneity

among user organisations. While it is desirable to cater to the specific needs of each user, GES vendors cannot emphasize particularities across all implementation sites. Accordingly, some IS literature focus on how GES vendors seek to gain overview of this variety, and how they work to align users where needs are incompatible.

In the process of informing generic design, gaining an overview of varying needs for GES vendors is deemed challenging. Some aspects of design can follow universal principles (Norman, 2013; Grudin, 1992). However, designing a one-size-fits-all cannot be achieved due to factors such as diverse practices, cultures and terminology existing within a large and heterogeneous audience of user organisations. (Soh et al., 2000). Outcomes of these challenges have been exemplified within existing IS literature, focusing on how generic features of the software clash with idiosyncratic needs of the individual user organisations, affecting existing practices (Strong & Volkoff, 2010; Lyytinen et al., 2016; Davenport, 1998;). While yielding useful insight into the understanding of the fit between generic systems and organizational routines, literature has mainly addressed the challenge of varying needs at the moment of implementation. Kallinikos (2004) argues that this focus has a rather unambiguous prescriptive orientation; *"they are by and large concerned with depicting the factors that may inhibit the successful implementation of ERP systems…"* (2004, p. 13). In line with Kalinikos (2004), Pollock, Williams and D'Adderio (2007) argue that the focus on implementation draws attention away from understanding the process of what informs design of generic software (2007). As such, they shift the debate from understanding how technology is made to work within particularities to how they are built to work across a diverse range of settings (Pollock et al., 2007, p. 257). They argue that generic systems are not designed from 'nowhere', such as some earlier research suggested (Bansler & Havn, 1996). Rather, design of generic software was based on input from relevant user groups, and informed through a process of generification. The concept describes a set of mechanisms used to gain a coherent understanding for design where specific needs are lifted out of their context and abstracted into 'generic' requirements. This is achieved through a process of negotiation where the goal is aligning needs between a wide set of user groups (Pollock et al., 2007).

The concept of generification has gained some traction in describing how vendors gain a coherent understanding for informing design. Johannesen & Ellingsen describes how generification strategies were used to inform design when taking an existing bespoke system and making it into a generic package (2009). Other studies have also discussed and exemplified how generification processes have contributed to informing design and development of a health information software (Fruijtier & Pinard, 2017; Nielsen, Bygstad & Gizaw, 2017). Some variations on the concept of generification exist. However, the commonality is describing how vendors gain a coherent understanding for design through some form of negotiation and dialogue between user groups, where the focus is on alignment of needs. In line with Kalikos (2004) and Pollock et al (2007), we argue that the large focus on implementation is drawing attention away from understanding the context of design and development that is shaping a GES. There is a lack in literature exploring how vendors design and develop a GES, which we argue is in need of a bigger focus. In addition, we have one concern with existing literature. Specifically, we question whether a sole focus on alignment and generification might make it challenging to capture the nuances of what goes into gaining a coherent understanding for informing design in this context, hindering a vocabulary that truly reflects the real life processes.

3 Research approach

The empirical basis for this paper is data collected through an ongoing Action Research (Davison et al., 2004) project where we collaborate with the vendor of the generic enterprise software 'DHIS2'. We will explain some details about the software and a set of key actors before elaborating on our methods for data collection and analysis.

3.1 Case description

Our empirical basis is a study of the design practices within a global action research project, named the Health Information Systems Programme (HISP). Central to the project is the development of a generic enterprise software, named "DHIS2". The District Health Information Software 2 (DHIS2) is designed to support the gathering and use of health information. The system is a free and open source software, implemented on a global scale in more than 70 countries. The software was initially built to support low and middle-income countries, but today DHIS2 also supports non-governmental organisations, for usecases such as disease surveillance of COVID-19, and health information management in high-income countries.

Our research focuses on four key actors. The first being the vendor, referred to as the core team. The core team is located at the University of Oslo and is mainly in charge of developing the generic applications of the software. The core team is made up of 80 staff members, a relatively modest size compared to the 2.4 billion people who live in the countries where DHIS2 is used. The internal structure of the core team is organized in product teams, each representing a set of core applications in the DHIS2 software. Through a variety of practices each product team gathers information and requirements coming from different user-organisations and other key actors. Requirements are fed into a ticketing system, named "Jira", in which product teams use to coordinate and plan their development activities.

The second category is the HISP nodes located in several countries around the world. The HISP nodes are consultancy firms, specializing in implementing DHIS2 for user-organisations. The third group of actors is user-organisations, such as ministries of health and private organisations that use the software as part of their information systems. Lastly, the fourth group is donors, mainly represented by Nongovernmental organisations. The donors are both users of the software and important financial partners supporting the vendor in developing the software. The HISP nodes, user-organisations and donors are collectively referred to as the "global community" by the core team. Table 1 summarizes the key actors.

Key actors	Description
Core team	Vendor of the software,located at the University of Oslo.
HISP Nodes	Consultancy firms, specializing in implementing DHIS2 for user organisations. Located in countries around the world such as Tanzania, Vietnam, India and Uganda.
Donors	Mainly Non-governmental organisations. Both financial partners and users of the software
User organisations	Ministries of Health, private health organisations, clinicians and other administrative health organisations

Table 1. Overview of key actors.

3.2 Data collection and analysis

The empirical data results from the diagnostic phase of our Action Research project. One of the authors has been involved in collaborative research with the vendor and other actors related to the

DHIS2 software for several years. This includes diagnostic and interventionist research focusing on design and innovation practices. The last six months a specific Action Research-project has been initiated in collaboration with the vendor, including all authors of this paper. The aim of the project is to diagnose challenges related to how the vendor works to gain a coherent understanding of a diverse user base, and to plan, implement, and evaluate interventions to address these challenges. The diagnosis has thus far included four in-depth interviews with product managers from the core team. Further, a document analysis of various online material relating to requirements gathering has been conducted. The interviews were recorded, transcribed, coded and themed in three rounds using a general inductive approach (Thomas, 2006). The approach structures the analysis in three steps; (1) Condensing raw data into summary formats, (2) establishing relationships between research objectives and summary findings and (3) developing categories into abstract concepts capturing key themes and processes.

We started out by condensing the raw data based on activities used to inform generic design, including what information is collected in each activity and from where the information was gathered. The condensed information was then further processed into brief summaries of the different approaches and practices that inform generic design. The brief summaries of key approaches helped us move to the next step of our analysis, establishing links between the findings and research objective. In the second step, central themes from each approach that contributed to informing the generic design of DHIS2 was identified. The themes were identified based on analysing how different approaches relate, as well as the concept of generification from the literature. Through this categorisation, traits used for a final comparison and analysis emerged. Finally, we developed abstract concepts capturing all underlying processes found in the empirical data. Additionally, to ensure the outcome of our data analysis correlated with the reported conditions, we continuously discussed our findings with representatives from the vendor. This will in the future be used to inform further interventions.

4 Findings

We now turn to our findings, where we examine how the vendor strives to gain a coherent understanding for informing the design of DHIS2. The global community represents a wide range of diverse needs and contexts of use. The core team is daily faced with the challenge of developing features that sufficiently meets the needs from this diverse set of actors. This has been increasingly difficult, as the DHIS2 software has grown exponentially in use of the last decade. Former ways of informing the generic design was based on developers from the University of Oslo travelling to countries, gathering requirements and learning, as well as informal experts users being able to send requirements to the core team by email. However, as the global community of DHIS2 grew with the implementation of the software in new countries, these more informal systems posed challenges. As explained by a product manager: "*It became problematic because of the lack of structure on our end to be able to absorb and respond to these various community requests became a bottleneck".* Due to the increasingly difficult challenge of managing and satisfying the diverse needs, the core team needed to experiment with new ways of informing their design. We will now elaborate further on these practices. The practices can be categorized into three approaches; i) Aligning needs amongst actors ii) Direct engagement with userorganisations iii) Monitoring of the global community.

4.1 Aligning needs amongst actors

With the previous informal structure used for gathering requirements, the core team experienced a bottleneck when handling requests from the community. The almost impossible task of extracting useful information from the large number of user needs became too tedious for the small core team. To streamline this process, three new practices emerged, where aligning needs between the actors became the focal point. The three practices were i) Outsourcing alignment work to expert users ii) Aligning

needs in a global requirement ticketing system and iii) Aligning user needs in feedback sessions. A common trait among these practices is that they inform design through a process of discussion and alignment activities between the core team and the members of the global community. The alignment activities are leveraging the core team by reducing the time spent on administering large amounts of individual requests, resulting in a more efficient system for handling requirements and solving the previous issues of information overflow. While all the practices share a commonality in alignment, they deviate with individual characteristics which we will elaborate on further.

4.1.1 Outsourcing alignment work to expert users

In order to overcome the issues related to extensive amounts of singular feature requests, the core team started to utilise the resources available in the HISP community. They appointed community representatives from the HISP groups to serve on a board of "field ambassadors". A representative from the core team described the role like this: *"They are communicating with all the field groups and coming up with their prioritized list of features or requirements. And then they feed that one list back to me"*. Essentially, field ambassador work relates to conveying relevant information from the field over to the core team, by filtering out requirements that are too particular and aligning those that are suitable for the general design of the system. The core team continues the alignment process by finding the commonalities among the all of the aggregated lists received from each field ambassador, which ultimately becomes a part of the future plans for the system.

4.1.2 Aligning needs in global ticketing system

The field ambassador system has its limitation of being a closed off loop between the field ambassador and the context in which it operates. Each field ambassador may strategically choose its sources of information and there is little clarity in what type of requirements that ends up going to the core team. To solve this, the core team utilised Jira as not only a coordinating system, but also as a global ticketing system that allows any actor within the network to request changes, report bug fixes or address the need for new functionality. This opened up the requirements handling process as not only field ambassadors could inquire directly with the UiO core team. It also made the overall process more transparent as it made actors in the DHIS2 community aware of what features were already being requested. The openness enabled the UiO core team to align needs amongst a broad set of actors as they could find commonalities among the many feature requests occurring on the site. Alignment within the community did also occur through comments on Jira:

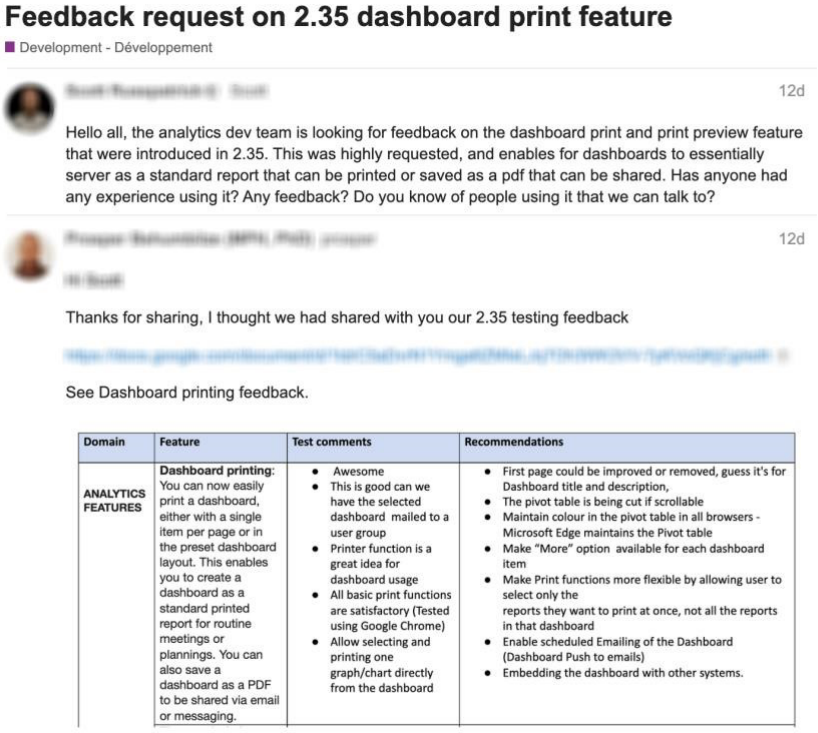
"Usually within the Jira tickets you see a lot of dialogue between the people who are in the field, hisp groups, power users etc. refining and updating the requirements." - Informant from the core team

By utilising the commenting feature existing on Jira, the core team could align concerns or gain a deeper understanding of needs related to a specific feature. The back and forth communication as well as the openness to the community makes Jira central to aligning needs.

4.1.3 Aligning user needs in feedback sessions

Alignment work is not only related to gathering requirements from various users, it is also central when receiving feedback on functionality that is already in production. When new features are to be created, the core team designs and presents mockups of the planned functionality to align different needs before it becomes a part of the core functionality. Aligning feedback ensures that the features

produced are covering the most central use cases that affect the largest part of the community. Aligning user feedback occurs in two ways; in weekly meetings between the core team and the field ambassadors and by posting design mockups on the global discussion forum for DHIS2. The first being a negotiation between experts users and the latter resulting in a wide variety of feedback from a broad set of actors. Image 1 illustrates how user feedback occurs on the global discussion forum for DHIS2.



Feedback request on 2.35 dashboard print feature

Development - Développement

12d

Hello all, the analytics dev team is looking for feedback on the dashboard print and print preview feature that were introduced in 2.35. This was highly requested, and enables for dashboards to essentially server as a standard report that can be printed or saved as a pdf that can be shared. Has anyone had any experience using it? Any feedback? Do you know of people using it that we can talk to?

12d

Thanks for sharing, I thought we had shared with you our 2.35 testing feedback

See Dashboard printing feedback.

Domain	Feature	Test comments	Recommendations
ANALYTICS FEATURES	**Dashboard printing**: You can now easily print a dashboard, either with a single item per page or in the preset dashboard layout. This enables you to create a dashboard as a standard printed report for routine meetings or plannings. You can also save a dashboard as a PDF to be shared via email or messaging.	• Awesome • This is good can we have the selected dashboard mailed to a user group • Printer function is a great idea for dashboard usage • All basic print functions are satisfactory (Tested using Google Chrome) • Allow selecting and printing one graph/chart directly from the dashboard	• First page could be improved or removed, guess it's for Dashboard title and description, • The pivot table is being cut if scrollable • Maintain colour in the pivot table in all browsers - Microsoft Edge maintains the Pivot table • Make "More" option available for each dashboard item • Make Print functions more flexible by allowing user to select only the reports they want to print at once, not all the reports in that dashboard • Enable scheduled Emailing of the Dashboard (Dashboard Push to emails) • Embedding the dashboard with other systems.

Image 1. Screenshot from Community of practice for DHIS2.

4.2 Direct engagement with user organisations

While the focus on more "generic" requirements has been necessary in order to manage the amount of incoming requests, the lack of contact with the actual contexts of use removed important granular details used to inform the design. In actively trying to combat this, the core team has a set of practices focusing on direct engagement with user-organisations. Here, the DHIS2 core team works together with a single organisation, gathering more granular requirements that feeds into the process of informing the generic design. This form of engagement is organized in two ways: i) Retrieving feedback and requirements directly from user-organisations and ii) Cooperating with donors on solving more specific-use cases. While different in scope and focus, the overall goal of this approach is informing generic design through learning and gathering more granular needs directly from the context in which DHIS2 is implemented.

4.2.1 Gathering requirements directly from a group of user organisations

The field ambassador system' function is to convey relevant information from the field, ensuring that the core team keeps in contact with the end-users of the software. However, a challenge with this practice is that the field ambassadors themselves are often several layers removed from the actual context of use. One informant describing the challenge said;

"*They are as much of a tourist if they go out to the community health workers, as I would be. You know they are just hanging out in the cities and getting requirements but they are not actually seeing what the people are doing in field hospitals."*

In response to this, the core team established a direct stream of communication with a selected group of local health workers in rural Rwanda. Here, the core team is directly engaging in a Whatsapp messaging group where people from HISP Rwanda can raise their concerns and share their knowledge directly to a product manager. The information gathered is unfiltered and not a product of an alignment process between multiple actors. Rather, it is based on pictures and messages of their day-to-day work, where they address some of their key insights as well as their everyday struggles. This direct way of communicating acts as a case study, giving the core team an unfiltered view of what the user needs might be, allowing them to learn and gather key insights in the process.

Rwanda being the country chosen for this practice was not accidental, but rather strategically chosen based on their relevance and qualities. As one informant described; *"First of all, HISP Rwanda is very organized. Rwanda as a country has a lot of qualities that we would like to reinforce with other nations as well".* Gathering requirements directly from the use-contexts like Rwanda is valuable for the core team. However, due to the limited size and capacity of the core team itself, the scalability of the practice is limited. Therefore, this practice is supplemented with donor cooperation.

4.2.2 Cooperating with donors on solving specific-use cases

NGO´s are the main donors for DHIS2, but also important users of the software itself. With a global presence, they often have a finger on the pulse in terms of what needs to be prioritised, representing an invaluable stream of information to the core team. Therefore, the core team sometimes engages with the donors, building out specific functionalities that they need. However, the main goal of DHIS2 is supporting local ministries of health, and not necessarily well funded organisations like donors. Thus, the core team also uses these projects as learning platforms to inform the design of DHIS2´s generic functionality. One example of this explained by an informant was;

"[Donors name] is using Tracker in 100 countries, and they had a specific use case. So we did a 3month process with them implementing this, while at the same time assessing how we could make it generic."

4.3 Monitoring of community activity

The user base of DHIS2 is both large and spread out over many different countries. In attempting to gain an overview of the needs of the broader community, without using large amounts of resources, one important stream of information is continuous monitoring of community activity. Being less resource intensive than direct engagement with singular organisations, monitoring global communication channels offers a possibility to easily gather a wide range of information that continuously informs the design of their generic functionality. Monitoring of community activities mainly happens on two platforms: Community of practice site (COP) and the DHIS2 App Hub.

4.3.1 Community of practice and App Hub

The community of practice is a globally available channel, where actors from the ecosystem engage in different topics of discussion. The channel has a lot of engagement, being generally useful to the core team as they can gather information on important topics of discussion, questions that need to be ad-

dressed and requests of improvement to different aspects. As one informant explained the purpose: "*The COP is the most raw, uncurated information source. It gives us a real perspective and understanding of the diversity in the global community".*

The App Hub is equivalent to the "App store" or "Google Play" and consists of developed apps made by third party actors in the various contexts of implementation. One product manager explained the importance of monitoring this channel, saying *"Most of the apps developed we have no idea of it happening, we only find out about them when someone posts them to the App Hub".* As these 3rd party innovations often represent enhancements to core functionality that is of importance to a set of users, they might provide inspiration or information that can be extracted to further inform the process of designing the core functionality.

5 Discussion

In this paper we aimed at answering; "*How can a vendor gain a coherent understanding of a diverse user base for informing generic design?"*. In addition, we listed a concern regarding the existing literature. We questioned whether the sole focus on alignment and generification in the literature was able to capture all elements of how vendors inform generic design. First, we will discuss our research question by focusing on the three approaches identified above. We have examined the role they play and will discuss how they contribute to the overall goal of gaining a coherent understanding for informing design. Then, we discuss the three approaches in relation to our listed concern with existing research.

5.1 Three approaches for informing design

Table 2 highlights the different approaches and their primary role in informing design. Each approach differs in what role they play for the vendor in gaining a coherent understanding for informing generic design.

Primary role in informing design	Approach to inform generic design	Examples of practices
Align user needs	Practices inform design through a process generification work	Outsourcing alignment work to expert user Aligning needs in global ticketing system
Gain insights into specific practices and needs	Direct engagement with user organisations, gathering more granular needs directly from the context in which software is implemented	Gathering requirements directly from a group of user organisations
Gain overview of the needs of the broader community	Providing global communication channels and monitoring them to gain perspective on diversity within user groups	Monitoring global communication channels

Table 2. Overview of the different approaches the vendor takes in striving to gain a coherent understanding for informing generic design.

First, the set of practices focused on aligning needs amongst actors supports the vendor in gaining a coherent understanding by contributing to finding commonalities in a diverse user base. This is in line with how Pollock et al. (2007) describes the process of generification work, as the overall goal of aligning user needs is the same. Similar to generification, the set of practices aid the prioritization of requirements by highlighting the most prominent needs, making the vendor aware of what requirements are in most demand. Existing literature has mainly described generification work as being conducted in 'alignment workshops', where a chosen set of user-representatives engage in negotiation and align needs (Pollock et al., 2007). Similar practices are present in our case, with the use of appointed expert users to conduct alignment work. However, we also see that generification work is conducted at a more public level through the global ticketing system. The openness of the system invites not only a selected few, but everyone within the global community of users to voice their opinion and engage in alignment work.

Second, the approach of direct engagement contributes to a coherent understanding through gaining insights into specific practices and needs directly from the context in which the software is implemented. This is similar to traditional approaches of design and development, where a granular understanding of user needs and end-user involvement are an important focus of informing design (Norman, 2013). Adhering to the idiosyncratic needs of each individual user organization is challenging for GES vendors (Strong & Volkoff, 2010; Soh et al., 2000). However, we see that the vendor tries to gain rich insights as it serves as an important learning platform for informing the design of generic features. Our case illustrates this importance by demonstrating how direct engagement with a user organisation in Rwanda provides the core team with an unfiltered view of how the software fits with their day-to-day work. This direct line of communication provides the core team with more granular details on what the everyday struggles with the software is, and what the needs might be. While the alignment approach gathers requirements from several user organisations at the same time, the information informing these requirements often travels through several levels before reaching its final destination; the core team. Elevating information through several alignment processes removes the contextual particularity of each requirement, resulting in a loss in granular details. Direct engagements removes the hierarchy between the vendor and the actors working in local contexts allowing the vendor more detailed insight into how the software is actually received by end-users.

Finally, the third approach adds to a coherent understanding of the divided user base by allowing a vendor to gain an overview of the needs of the broader community. Contrary to a focus on aligning needs, the role of this approach is to get a broader sense of what type of needs are occurring within the diverse set of users. This differs from approaches reported in existing research, where keeping a certain 'distance' from users and mainly interacting with a small selection of user organisations is discussed as necessary when designing generic software (Pollock, Williams & Procter, 2003; Koch, 2007). However, shying away from focusing on a broader perspective of user needs could create a situation of imbalance where the particular needs of a selected few gains too much foothold (Wagner, Scott and Galliers, 2006). The approach of monitoring the community activity addresses this is by ensuring that the software does become too specific to a few user organisations. One example being donors potentially having supremacy over other user organisations due to being the monetary provider for the system. By actively monitoring the whole community, the core team is continuously given a perspective on the diversity of needs. This knowledge is further utilised in creating software capable of catering for a diversity of needs across geographically and culturally divided implementation sites.

5.2 The role of generification

Through our case we have shown how generification is a central approach to gaining a coherent understanding and informing generic design. Extant literature conceptualises generification as the practices of making software generic through a focus on aligning needs (Pollock et al., 2007; Johansen & Ellingsen, 2009). However, our case also illustrates two other approaches with different primary roles in gaining a coherent understanding for informing generic design. While each approach plays different

roles in informing design of generic features, a GES vendor will more often than not try to include features needed by many and neglect features relevant to one or few. In order to do this, some form of alignment between diverse needs is ultimately needed. Therefore, one could say that the role of monitoring and direct engagement is informing the process of generification by providing supplementary perspectives, and it is the alignment of these perspectives that ultimately helps the vendor in gaining a coherent understanding for informing generic design. Thus, one might view the approaches of monitoring and direct engagement as strategies of generification work, as they ultimately contribute to the same end goal of creating generic design based on aligned needs. On the contrary, Pollock et al. originally emphasized generification work as not only a method to align user needs to inform generic design, it is also used as a means to *"[...] shape the users attitudes toward the overall generification process"* (2007, 263) and as a *"method [...] of moving users towards the 'organizationally generic.'"* (2007, p. 269). In other words, Pollock et al. (2007) accentuate generification as a two-sided social activity, where both the vendor and the acainted actors become aware of their similarities and differences. We argue the approach of aligning user needs are illustrating this bilateral engagement and therefore fit the given original conceptualization. The two remaining approaches described do not encompass the same reciprocal understanding between vendor and users, thus being outside Pollock et al. (2007) conceptualisation of generification.

In summary, whether the three approaches identified are categorised as generification strategies or not, their distinctive roles remain the same in informing generic design. In addition to the focus on alignment expressed by prior research, we argue that a vendor can gain a coherent understanding by utilising a set of approaches focused on covering two additional roles. These are; i) Gaining an overview of the variety of needs among user organisations ii) Gaining insights into the details of the work practices they aim to support. We see that focusing on the three roles combined are what supports a GES vendor in gaining a coherent understanding of a diverse set of users.

6 Conclusion and further research

In this paper, we explored the question; *How can a vendor gain a coherent understanding of a diverse user base for informing generic design?* In our case, the vendor had three approaches, each playing different roles for gaining a coherent understanding. The three approaches and their roles were; i) A set of practices focused on aligning needs amongst actors ii) the approach of direct engagement focusing on gaining insights into specific practices and needs iii) Monitoring the community to gain an overview of the needs of the broad user base. These approaches give an insight into how a vendor operates within the context of designing a GES, a context otherwise little explored by IS research. Extant literature has had a sole focus on alignment as mean to gain a coherent understanding for informing generic design. We see that the two additional approaches identified in our case also play important roles for vendors to achieve a coherent understanding of a diverse user base.

While the specifics of each approach identified are limited to our case, an avenue for further research would be to look at other GES and examine how the role of each approach apply to other cases. It would also be interesting to examine if other approaches and roles exist to help the vendor gain a coherent understanding, further uncovering the context of designing a generic enterprise software.

References

Bansler, J. & Erling, H. (1996) 'Industrialised Information Systems Development', *CTI Working Paper* (22.), Technical University of Denmark.

Baxter, G., & Sommerville, I. (2011). Socio-technical systems: From design methods to systems engineering. Interacting with computers, 23(1), 4-17.

Cornford, J. and Pollock, N. (2003) Putting the University Online: Information, Technology and Organisational Change. Milton Keynes: Open University Press.

Davenport, H. (1998). Putting the Enterprise into the Enterprise System. *Harvard Business Review,* 1998(July), 121-131.

Davidson, R., Martinsons, M., Kock, N. (2004). Principles of canonical action research. *Information Systems Journal* 14(1). 65-86.

Dittrich, Y. (2014). Software engineering beyond the project–Sustaining software ecosystems. Information and Software Technology, 56(11), 1436-1456.

Fruijtier, E. & Pinard, M. (2017). Collaborative development of Global Information systems: Toward Community based Generification. *EJISDC* (2017), 78, 3, 1-20.

Gizaw, A, Nielssen, P. and Bendik, B. (2017). Open Generification. *Information Systems Journal* (2017) 27, 619–642.

Gregory, J. (2003). Scandinavian Approaches to Participatory Design. *Int. J. Engng Ed.* 19(1), 62-74.

Grudin, J. (1992). Utility and usability: research issues and development contexts. Interacting with computers, 4(2), 209-217.

Johannessen, L. & Ellingsen, G. (2009) Integration and Generification: Agile Software Development in the Healthcare Market. *Computer Supported Cooperative Work* 18(5), 607-634.

Kallinikos, J. (2004). Deconstructing information packages: organizational and behavioural implications of ERP systems. Information Technology and People, 17(1), 8–30.

Koch, C. (2007). ERP: a moving target. *Int. J. Business Information Systems*, 2(4), 426–443.

Lyytinen, K., Nicholas, B., Yoo, Y., King, J.L. (2016) Routines as Shock Absorbers During Organizational Transformation: Integration, Control, and NASA's Enterprise Information System. *Organization Science,* 27(3), 551-572.

Norman, D. (2013). The design of everyday things: Revised and expanded edition: Constellation. Pollock, N., Williams, R. and Procter, R. (2003) 'Fitting standard software packages to nonstandard organisations: the 'biography' of an enterprise-wide system', *Technology Analysis & Strategic Management*, 15, 317–332.

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.

Rosson, M. B., & Carroll, J. M. (2009). Scenario-based design. In Human-computer interaction (pp. 161-180): CRC Press.

Soh, C., Kien, S. S., & Tay-Yap, J. (2000). Cultural fits and misfits: is ERP a universal solution? *Communications of the ACM*, 43(4), 47-47.

Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., . . . Paige, R. (2012). Large-scale complex IT systems. Communications of the ACM, 55(7), 71-77.

Strong, D. & Volkoff, O. (2010). Understanding Organization-Enterprise System Fit: A Path to Theorizing the Information Technology Artifact. *MIS Quarterly,* 34. 731-756.

Thomas, D. (2006). A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation*, 27, 237-246.

Wagner, E., Scott, S. & Galliers, R. (2006). The creation of 'best practice' software: Myth, reality and ethics. *Information and Organization,* 16, 251–275.

Wilkinson, C. & De Angeli, A. (2014). Applying user centred and participatory design approaches to commercial product development. *Design Studies,* 35(2014), 614-631.