NTNU
Det skapende universitet

Helseinformasjonssystemer i utviklingsland

Bruk av DHIS2 i en mHealth løsning i Zambia

Jan Alexander S Bremnes

Master i informatikk
Innlevert: desember 2015
Hovedveileder: Eric Monteiro, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

Summary

This thesis focuses on the topic of health information systems in developing countries and the challenges one faces when working in a low resource context. It is a case study with an approach based on research principles from Action Design Research where the case in focus is the configuration and implementation of DHIS2 as a mobile to web solution for an mHealth intervention in the field of iCCM in Zambia. I stayed in Zambia for about three months, from end of May to mid-August 2015 and during this period worked closely together with two locally based NGOs, Akros and ZCAHRD. This intervention is part of a larger study conducted by ZCAHRD and Boston University, with funding from UNICEF, that aims to evaluate the potential benefits of using community health workers using mobile phones for inventory management and to report disease statistics, as well as receiving feedback from the health clinics they are attached to. The solution developed went live in November, after having oveshot its original deadline with a few weeks, but the study is scheduled to last until June 2016 so a short delay should have no great impact on the results. This thesis discusses some of the problems that were encountered during development, and they are found mostly to be the result of a relatively inexperienced team working together in a highly distributed manner, as well as some unforeseen problems with funding and the national powergrid. Basing the solution on the free and open source software DHIS2, and working together with employees of the HISP network, allowed us to implement custom features that were not previously available in the software.

Sammendrag

Denne oppgaven fokuserer på temaet ”Helseinformasjonssystemer i utviklingsland” og de utfordringene en kan møte på under arbeid i en lavressurskontekst. Oppgaven er en case study som baserer seg på forskningsprinsipper fra Action Design Research og den spesifikke casen er arbeidet med å konfigurere og implementere DHIS2 for bruk i en mobil-til-web løsning tilknyttet feltet iCCM i Zambia. Under arbeidet med denne oppgaven tilbragte jeg omtrent tre måneder i Zambia, fra slutten av Mai til midt i August 2015, og i denne perioden jobbet jeg tett sammen med to lokalt baserte organisasjoner, Akros og ZCAHRD. Arbeidet med denne løsningen for iCCM er en del av en større studie som foretas av ZCAHRD i samarbeid med Boston University og er finansiert av UNICEF. Denne studien akter å evaluere de potensielle fordelene ved å la lokale helsearbeidere bruke mobiltelefoner til å bestille medikamenter, innrapportere sykdomsstatistikk og motta tilbakemelding fra de lokale helseklinikkene de er tilknyttet. Løsningen ble tatt i bruk i November, noen uker bak skjema, men ettersom den er tenkt å vare til Juni 2016 burde ikke forsinkelsen ha for mye å si på sluttresultatet. Oppgaven diskuterer problemene som dukket opp under utviklingsarbeidet, og de blir vurdert til å hovedsaklig skyldes et relativt uerfarent team som jobbet sammen på en distribuert måte, i tillegg til noen uforutsette finansielle problemer og mangler i det nasjonale strømnettet. Ved å basere løsningen på DHIS2, som er fri og åpen programvare, og ved å jobbe sammen med ansatte i HISP-nettverket, fikk vi muligheten til å implementere skreddersydd funksjonalitet som til da ikke hadde eksistert i DHIS2.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACT	Artemisinin-based Combination Therapy
BU	Boston University
CC	Community Champion
CHA	Community Health Assistant
CHAP	Community Health Access Point
CHW	Community Health Worker
CLTS	Community Led Total Sanitation
DC	Developing Country
FOSS	Free and Open Source Software
GHC	Global Health Corp
GSMA	GSM Association
GSM	Global System for Mobile Communications
HC	Health Center
HP	Health Post
iCCM	Integrated Community Case Management
ICT4D	Information and Communication Technology For Development
ICT	Information and Communication Technology
IPR	Intellectual Property Rights
ISDC	Information Systems in Developing Countries

J2ME Java 2 Platform, Micro Edition

MCDMCH Ministry of Community Development, Mother and Child Health

MDG Millenium Development Goals

MoH Ministry of Health

NGO Non-governmental Organisation

ORS Oral Rehydration Salts

OSS Open Source Software

RDT Rapid Diagnostic Test

SMS Short Messaging Service

TCO Total Cost of Ownership

WASH Water, Sanitation and Hygiene

ZCAHRD Zambia Center for Applied Health Research and Development

Chapter 1

Introduction

1.1 Motivation

Developing countries face immense challenges in the public health domain, due to a multitude of infectious diseases which have been eradicated in the richer parts of the world, such as malaria, and periodic outbreaks of terrible diseases with no known cure, such as the recent devestating outbreak of ebola in Western Africa. Developing countries often have large rural populations that have poor access to proper health care, if any, and easily preventable diseases that cause too many deaths is all too common because of poor sanitation and hygiene.

In order to fight disease and provide proper healthcare for the general population, one needs the resources to do so. Not only of the financial kind but also resources such as statistics on population and disease outbreaks. But many developing countries are still stuck with a paper based system were data collected often end up in a dark closet or filing cabinet and rarely see any use.

The steadily decreasing cost of, and increasing access to, computing power and computing devices such as laptops and mobile phones provide an opportunity for developing countries to adopt a more modern approach.

Donor agencies and non governmental organisations are putting time, effort and money into development projects based on modern information and communication technology, and several open source software tools for collection of health data statistics and patient registration exists, such as openMRS and DHIS2. Modern web browsers, which are able to run feature rich web-based applications, together with ever cheaper and more capable mobile devices and an increasing internet coverage allows developing countries to move from paper based records tucked away in dark closets to a advanced computer systems allowing real-time data capture and analysis of national health statistics. The use of mobile devices such as tablets, with powerful processors, mobile internet connectivity and GPS allow health workers to track and map disease outbreaks in order to better combat them. There is great interest from both industry and organisations such as UNICEF in the application of mobile-based solutions to provide not only better health care, but also improvement of reading skills and banking services for the poor in developing countries.

But the work and challenges involved in implementing and developing information systems for use in developing countries is not always the same as when working in the developed part of the world. Researchers and developers who come from Western countries may be unfamiliar with the challenges involved with working in a low resource context, where everything from electricity to spare parts and manpower can be in short supply and the political climate can be unstable. While one can always get an impression of what it is like to work under circumstances that differ greatly from what one is used to by reading the literature and thereby hopefully learning from the mistakes of others, it is no substitute for actually going into the field and experience first hand the challenges, and rewards, involved.

1.2 Research Objectives

The broader theme I want to study is how information technology can be used for development but more specifically, I wish to look at implementation of an information system in a low resource context and the challenges that may present.

- Does free and open source software provide any benefit when working in a resource constrained environment?
- Can software development methodologies be applied to the planning and development of an information system in a low resource context?

As part of my work, I spent approximately three months in Zambia, where for most of the time I worked in close cooperation with two locally based NGOs, Akros and ZC-AHRD, helping them set up and configure DHIS2 for the m-Health intervention. This intervention is part of a larger study on the effectiveness of integrated community case management (iCCM) conducted by Boston University with support from UNICEF. Community health workers in rural Zambia is given training in how to diagnose and treat a handful of diseases that are the main killers of children under the age of five. They are given small stocks of medicine and supplies every month, and are the primary health care supplier in the areas they work. The iCCM study is intended to identify areas of improvement in both the care giving and the inventory management.

The DHIS2 software is developed by the Health Information Systems Programme (HISP), a global network established and coordinated by the Department of Informatics at the University of Oslo in Norway, and this thesis is written in cooperation with them as the final part of my Masters programme at the Norwegian University of Science and Technology.

1.3 Personal Motivation

I consider computer science and information technology exciting and interesting in and of itself, but it has never been my primary interest. Instead of studying the technology itself, I find it more interesting how technology can be used to improve the world we live in. Any technology in itself is worthless, it is through the applications of the technology that it shows its true potential. I had for a long time been somewhat annoyed and frustrated by what I viewed as a lack of ambition and [and what?] in my field of study. If you are an electrical or mechanical engineer then there seems to be a lot of opportunities for you to put your skills and knowledge to use together with organisations such as Engineers Without Borders, and if you are a doctor or nurse you can work with Doctors Without Borders, the Red Cross or a number of other organisations. There didn't seem to be anything similar for anyone studying informatics. That is why, while looking through my University's suggested topics for master thesis, wondering what I should choose, the topic "Information systems in developing countries" immediately caught my eye.

In addition to providing me the opportunity to spend parts of my final year at the University actually doing something that hopefully might be immediately useful, instead of producing a thesis that could very well see the completion of my studies as its only purpose, I hoped it could also give me some valuable experiences and new perspectives.

Proper health care is severely lacking in many developing countries, and there are many reasons for this; from the lack of properly trained health workers to an abundance of infectious diseases. While information technology cannot help with all the different problems, it at least has the potential to improve certain areas of the health care system. Health information systems can improve the information flow so that hospitals can get patient records from rural clinics when they receive a patient instead of starting from scratch with diagnosing the patient. Mothers can get text messages reminding them of appointments for child vaccination, and clinic staff can log their visits and make sure they follow through with the entire vaccination program. These are all systems that developed countries have had in place for a long time, but they have been built up over many years for a particular country's specific needs.

Looking at our own healthcare system here in Norway, and in other developed countries, it is easy to view it as entrenched in bureaucracy and long established rules and regulations that impedes innovation, with any change or new system that is proposed having to go through a lengthy evaluation stage before seeing any chance of being implemented. Where the cost involved in any new IT-system can be enormous and the failures equally spectacular. There are good reasons for this of course, since a persons health data is seen as very sensitive and therefore there needs to be safeguards in place to protect it and prevent it from falling into the wrong hands. Developing countries can not necessarily spend the same amounts of time or money on their health care systems, so they need more different approach. Mobile phones have seen many interesting uses in developing countries, and I remember reading a story in Time Magazine about the innovations happening in Eastern Africa with low-tech mobile solutions providing banking and money transfer services for people who had never had access to a traditional banking account. I was left wondering how it could be that in rural Kenya two neighbours could settle a monetary debt more quickly and easily than I could. There seems to be room for a lot of exciting innovation in developing countries, since they have only recently started adopting modern information technology and do not have a set of preconceptions about how things should be done and see instead what can be done.

1.4 Structure of Thesis

Proceeding from here, the rest of this thesis is divided into three main parts. Part one provides some theory and bakground for this project and intends to show how information technology can be used in development work with a focus on the use of free and open source software and mobile technolgies and how these two phenomenon can enable populations in developing countries to enter the digital era. Following this, part two will give an introduction to HISP and the DHIS2 software before discussing the specific project that I worked on and present the research methods employed. Finally, part three provides a discussion of my observations during my work in Zambia and finish off with some conclusive remarks.

Part I

Background and Theory

Chapter 2

Using Information Technology for Development

In this chapter, I will take a look at the use of information and communication technology (ICT) in developing countries (DCs) and explore the concepts of the Digital Divide and the Learning Economy, after first giving a brief introduction to what is often called ICT4D and taking a look at some of the challenges that one might face if working in this field.

Information and communication technology (ICT) in developing countries (DCs) is a research area that did not see much activity until quite recently, but research activity has increased in later years and ISDC has matured more as a field of research (Heeks, 2002). While literature on the subject was previously found as sporadic articles in more general IS journals, there are now several journals dedicated to the topic, and leading journals in the field of information systems, like MISQ, have published special issues on the topic.

From the 1950s, when Kolkata, India saw the arrival of the country's first computer, the first time a digital computer was installed in a developing country, and until the 1990s, governments was the key actor and ICT was applied mainly to the administrative functions of the government in DCs. During the 1980s, multinationals and other large companies saw the opportunity for using ICT as a tool for delivering economic growth in the private sector. In the 1990s, the Internet and the World Wide Web [entered the mainstream public], and along with the Millennium Development Goals (MDGs), formalized by the United Nations, it sparked a generalized upsurge of interest in applying ICT in DCs. Development agencies and non-governmental organisations (NGOs) became the key actors and prioritised ICT as a way of realising these goals (Heeks, 2008). Note that I will frequently be using the term ICT4D throughout this thesis, which is an abbreviation of "Information and Communication Technology for Development". In the literature one can also find the term ISDC , for "Information Systems in Developing Countries", and these two are mostly overlapping.

2.1 Possible problems and challenges

Working with ICT in DCs brings with it a range of challenges and problems that one might not be used to if coming from a developed country, and being aware of these challenges and having a strategy to meet and overcome them can make all the difference for the possible success of the project. For instance, it is easy to take things like a stable electrical supply, easy access to spare parts and whatever hardware one needs, and fast and reliable internet connection for granted. And it is easy to forget that it is not like that in every country. DCs do notalways have a stable electrical supply, power outages may be common and long lasting, or electricity is rationed with planned powercuts every now and then, and Internet connectivity can be slow or non-existing and getting that vital computer part replaced may take weeks.

There can also be problems related to funding, though funding problems are probably common in most parts of the world, the difference is that while ICT-projects in the developed world is either undertaken by private businesses, and paid for by the business wanting a new system, or the local or national government, which pays for it with the taxpayers money, ICT-projects in DCs are often funded by external donors. Whether it is aid money from foreign governments, loans from the IMF, or projects run by NGOs, there's always the risk of the funds ending before the project is complete or allowed to mature.

It is sadly the fact that a lot of ICT projects fail; either partially or completely and be that in developing and developed countries. Heeks divides ICT projects into three groups:successful, partial failure and total failure (Heeks, 2002) and he states that there is evidence that shows a higher failure rate of ICT projects in developing countries than in developed countries. However, a study that compares the failure rate of ICT projects in the USA and South Africa over nearly a decade, finds that there is not that much of a difference in failure rates between the two countries, and that the percentage of successful projects has improved over time (Marnewick, 2012). While South Africa cannot necessarily be described as a developing country, its IT industry does not have the size and experience as its American counterpart, so it is not given that increased risk of failure is something inherent in ICT4D projects.

The failures that do exist, can often be attributed to problems concerning sustainability and scalability which are both very important factors to consider. Both have some relation to the fact that external organisations (NGOs, donor countries and similar) often are the ones driving these projects. The sustainability of a project is threatened if there is no work being done in order to ensure the continuation of the project after external funding and development is cut off. If a project does not have enough local support, in the form of money, skills and political will, it will have difficulties surviving after the external contributors pull out. With this in mind, it can be seen that training of local staff is an important factor, to make sure they have the skills and capacity to train new workers and users, as well as having the ability to deal with future problems that might occur. It is not sufficient to only have competent users if there is no one available to fix serious bugs that might crop up, or know how to restore a database in case of data corruption (Fraser and Blaya, 2010). Scalability can become a problem when NGOs have initiated a pilot project which is then handed over to local administration without anyone knowing if it is possible to scale the pilot project to a full implementation. It is not guaranteed that an ICT system that works well when handling aggregate data from a single village or town will work as expected when deployed at a national level.

All of this is merely to highlight the fact that there are risks and challenges involved in any ICT project, and that when working in DCs they may be different from what one is used to. Being aware of them and making an effort to avoid the more obvious pitfalls can increase the chance of success. However, there have been some concerns about whether ICT actually contributes to the development of DCs, or if it increases their dependency on the developed countries (Wade, 2002; Byrne and Jolliffe, 2007). We will come back to this concern in the next section where we examine how ICT can contribute to development in DCs.

2.2 Applying ICT for development

Through the years there have been some discussion about the relevance of ICT for DCs, and one can ask if ICT is beneficial to DCs or even needed when there are so many other, more fundamental, problems plaguing many of these countries. However, according to Walsham and Sahay, the "...debate has been resolved with a clear yes answer. The question has become not whether, but how ICT can be beneficial" (Walsham and Sahay, 2006). They also state that "[ICTs] have high potential value across all sectors in both public and private enterprises, and at multiple levels, from software businesses in urban areas, for example, to health delivery in rural villages."

There are several approaches to bringing ICT to DCs, ranging from the introduction of large scale information systems in public administration, to rural telekiosks supplying telephone and internet services to the poor. But it is not a question of one or the other, rather it is that both approaches are needed in order to successfully cross the digital divide. The introduction of large information systems will have a hard time being successful if there is no focus on enhancing the computer literacy of the users. As was mentioned previously, many ICT4D-projects are funded and initiated by external actors, either development agencies or NGOs, and they often fail because they did put enough work into building local capacity and skills to ensure the continuation of the project after the external actors leave and consider their work done.

According to Heeks there are "... two extremes along the continuum of different approaches to technology and development. At one end is the passive diffusion view. [This approach] holds that if ICT does have a developmental value for the poor, a combination of private firms' search for profit plus the poor's search for value will make it happen. [Conversely], the active innovation perspective feels the market will not deliver – or will deliver too slowly – to the poor. Hence, intervention is required in the form of innovations that will better help to meet development goals." (Heeks, 2008). It could be argued that the correct approach is probably somewhere in the middle. When trying to innovate on behalf of someone else, there is often many possibilities for error. There are many examples of external actors who see a problem in a DC and already have an IS that solves a similar problem in their country, and goes on implementing said IS in the DC believing that "if it works here, it should work there." This belief in the existence of "one size fits all" IS, which does not take into account the local conditions and the context in which it will be implemented, very often lead to failure (Heeks, 2002). Some argue that "Technology diffusion must be local user-focused rather than donor- or technology focused." (Davison et al., 2000).

This leads us to the concept of "design-actuality gaps" (Heeks, 2002) which can be described as the gap between the expected reality the system is designed for, and the actual reality that it faces during implementation and development. In order to increase the chances of a successful project, one should strive to minimize the design-actuality gap. Reducing the gap calls for extensive knowledge of the local environment as well as the ability for local improvisation when needed. Knowledge about the local environment can be gained by involving the people who are going to be the actual users of the system, or if that is not possible at least have someone connected to the development team examine the local environment in order to eliminate obvious design errors. Involving local stakeholders in both the design- and development process allows learning to take place on both sides, and as argued in (Ernst and Lundvall, 2004), learning is key to development.

2.3 The Digital Divide

One of the goals of ICT4D is to help DCs cross the so-called "digital divide", which is commonly "... understood to be a reference to classes of people at risk of being excluded from the rising tide of economic prosperity, fueled by great advances in information technology" (Kvasny and Truex, 2001) . Basically, it refers to the digital haves and have nots, and the divide does not only exist between different countries but also within countries. The rich, educated and urban population often have better access to ICT than the poor, uneducated or rural population and this divide exists to a varying degree in all countries, both rich and poor. The digital divide between countries mostly follows the divide between the developed and the developing world, and the underlying reasons are mostly economic and policy issues.

Being on the wrong side of the digital divide means that DCs are or severely restricted in their ability, or even unable, to participate in large parts of the global economy. However, crossing the divide is not only a question of access, but also that of ability. Access to modern ICT is of little use if you are not able to utilise it efficiently (Walsham and Sahay, 2006). Having access to computers and the internet is useful for just about everyone in developed countries, since it can be used for everything from booking an appointment with your doctor, to filing your taxes and looking up the phone number and address of the nearest pizza restaurant. We have a wide variety of things to use ICT for, which the general population in DCs do not, so they have fewer incentives to acquire not only access to, but the skills needed to use ICT. There have been attempts to give rural populations in DCs access to modern ICT such as computers, internet and telephone services, in the form of *telecenters*, also knowns as *telecottages* or *telekiosks*, which would be places people could go to get access to computing and communication facilities. While there were some enthusiasm about this in the beginning, a lot of failures in this area has caused a loss of faith in the approach and led some to dub telecenters as ”the rusting tractors for the 21st century.” (Heeks, 2005; Wade, 2002; Lucas, 2008).

While the developed countries has spent a long time getting where it is today, through a steady technological evolution and construction of the necessary infrastructure, the pace of technological advancement is so rapid that many argue DCs will not be able to catch up unless they undergo technological revolutions instead of a steady evolution. The term “leapfrogging” is often used in this context, which can be defined as “the implementation of a new and up-to-date technology in an application area in which at least the previous version of that technology has not been deployed.” (Davison et al., 2000). There are several examples of leapfrogging happening within DCs, and a common case is the introduction of mobile phones in areas or regions where landlines were practically non-existent outside cities and larger population areas.

There are both economical and ethical incentives for helping DCs cross the digital divide. Given that a large percentage of the worlds population live in DCs, there is a huge untapped market for companies in the ICT sector. ICT also has the potential, as we shall see later, to improve the lives and health of the billions of people living in DCs, as well as foster development and help them take part in the global economy (Avgerou, 2008).

2.4 The Learning Economy

“To put it bluntly, there is no way to reduce poverty other than to place learning and knowledge creation at the centre of the development strategy. Foreign aid and windfall profits from oil and other natural resources can produce sustained development only if these resources are channelled into the formation of local capabilities” (Ernst and Lundvall, 2004, p 266)

In (Ernst and Lundvall, 2004) the authors makes the case that learning and knowledge creation is paramount for the continued development of countries. One often hear talk about how the developed countries have left the industrial age and entered the information age, where “knowledge workers” are more highly regarded than the “unskilled”, manual labour workers. Capital investments across country borders is now larger than trade, so trade is no longer the main driver of global economic growth. DCs stand to gain from joining “international production networks”.

China is a country that has seen immense economic growth in recent decades, and this is commonly attributed to the country having become ”the worlds factory”, with giant manufacturing plants churning out products that are being exported around the globe. At first they were manufacturing products invented and designed by companies from other countries, many European and American companies outsourced large parts of their production to China, but more recently we have started seeing the emergence of Chinese companies as major global players, especially in the high-tech sector, e.g. Huawei, who is now a major competitor to Ericsson and Cisco in delivering network solutions. In a similar vein we have India, which is well known for having built a large domestic IT-sector, and many companies in Europe and the USA outsource many business functions and IT-related work to firms in India. However, there is little domestic IT-innovation, so the country does not benefit from it other than economically. This is in contrast to Brazil, which has a very small IT-sector, but what they have is quite innovative (Avgerou, 2008). Indian IT-companies are seeing competition from other DCs which offer lower prices. Since much of what the IT-firms do are mostly explicit knowledge, the field is open for a lot of competitors. For instance Rwanda, which is trying to reinvent themselves as a technology hub in Sub-Saharan Africa.

Before jumping to any conclusion that increased use of ICT is beneficial for DCs, one should consider the possibility that ICT is increasing DCs dependence on the developed countries, for instance if the adoption of ICT leads to reliance on proprietary software controlled by multinational corporations. As mentioned in (Ernst and Lundvall, 2004) “Access to codified knowledge may at times be constrained by patenting, aggressive IPR strategies and the proliferation of ’high-tech neomercantalism’”. International companies from developed countries are applying for software patents in DCs at an alarming rate, one study states that in the countries of Lesotho, Malawi and Mozambique only a handful of patents were filed by locals versus 150 000 by foreign individuals and corporations (Byrne and Jolliffe, 2007). This can work as a deterrent for local innovation and development since few people and companies have enough money to fight patent battles in court. Western companies can be very aggressive when it comes to pushing intellectual property (IP) rights and ”Armed with a self proclaimed belief that intellectual property rights are absolute, and supported by a barrage of tools of coercion ... private interests from developed countries are claiming property rights across Africa at an alarming rate.” (Byrne and Jolliffe, 2007).

There seems to be a need for an approach that encourages local capacity building and which facilitates learning and knowledge creation so that the DCs can strengthen their own position in the global economy by improving existing, or creating entirely new, industries. Or if not that, at least gain enough knowledge and skill to lessen their dependence on other countries. Encouraging an increased use of free and open source software, that does not put any restrictions on its use like proprietary software often does, might be one way to go about this. If we acknowledge one of the central ideas in the Free Software movement, that software should be free in order to make people more free, then this can serve as an argument in favor of FOSS if one agrees with Amartya Sen’s argument that freedom is the principal goal and purpose of developement (Sen, 1999). The next chapter will therefore focus on free and open source software.

Chapter 3

Free and Open Source Software

The previous chapter discussed the possible role of ICT contributing to the development of countries, and presented the ideas of leapfrogging and the digital divide. Crossing the digital divide requires both the means and the will to do it; people need access to ICT but they also need to have reason for using it. Reasons for using ICT may come naturally as people see the benefits and opportunity it provides, but what do you do if you have the need but not the means to gain access? This chapter will give a brief introduction to the world of free and open source software (FOSS). What it is, how it compares to proprietary software and present a few examples of widespread FOSS projects.

3.1 What is FOSS?

Free and Open Source Software is an umbrella term that encompasses all software that is released under an open source licensing scheme. A common misunderstanding is to believe that the ”free” in ”free and open source” means that the software is free of charge, or gratis. While it is often the case that FOSS is gratis to acquire, it is not necessarily without cost during deployment and use. Rather, the term ”free” should be understood in the sense of freedom, that it presents users of FOSS with the freedom to know what exactly their software is doing, and prevent it from doing things that they do not like. The term ”open” refers to the source code being open and available for anyone who is interested in having a look. FOSS allows users of the software to view, modify and often redistribute the source code with very few limitations. Compared to proprietary software, where you only get the compiled program and no access to inspect its inner workings.

It should be noted that there is a philosophical difference between free software and open source software, but I will use the commonly used term FOSS to cover both of them since the differences are mainly philosophical and not important for our discussion[1]. The difference lies mainly in that proponents of Open Source believes that source code should be public in order to increase the security and quality of the software by allowing everyone

[1] https://www.gnu.org/philosophy/open-source-misses-the-point.html

is interested to go through the code looking for bugs or areas of possible improvement. Those adhering to the Free Software movement, believe source code should be public and freely available so that the users can inspect the code running on their computers to make sure that their rights and freedoms are maintained, e.g. making sure that the music player application is not logging your keyboard strokes in order to steal passwords. None of the two camps is opposed to the idea of companies, or individuals, selling software for profit, they only argue that the source code should be made available as well as the binary executable.

3.2 FOSS vs Proprietary Software

There is no single development methodology that can be used to describe every FOSS project, but the open nature of FOSS means that the development often differs from how commercial proprietary software development works. Many FOSS projects are developed by a globally distributed team that is a mix of volunteers, hobbyists, employees at major corporations and sometimes paid staff attached directly to the project. Often, the source code is hosted at an online source code repository such as Github or Launchpad, and anyone who is interested can submit their code to the project. Most projects have one or more *maintainers*, which are the ones with control over the specific repository and decides which submissions are accepted and which are rejected. The maintainer is usually the person or company that started the project, or it might be someone they have entrusted with the responsibility of maintaining the project.

It used to be the case that FOSS projects were usually managed by individuals or a group of volunteers, e.g. the various Linux distributions, the Linux kernel, and the GNU toolchain, but increasingly large software companies contribute both manpower and financial resources to various FOSS projects, as well as releasing internal projects as FOSS. This shift in the nature of FOSS development, from distributed individual developers to companies and large organisations, has taken place over the course of several years and has led to the labeling of this new approach to open source development as OSS 2.0. Table 3.1 illustrates some of the differences between traditional FOSS and this newer development (Fitzgerald, 2006).

For most of the software the average person needs, there exists both proprietary commercial solutions as well as FOSS alternatives, and often there are several different offerings to choose from. There are plenty of web browsers freely available, and free operating systems such as Linux and FreeBSD compete with Windows from Microsoft and OS X from Apple, while LibreOffice offers much of the functionality provided by Microsoft Office, and so on. Common to most FOSS projects however, are that they are *horizontal* applications, which is often the result of ”itch scratching” and require no specialised domain knowledge outside of software development. Usually the developers are themselves users of the software they develop, and they started developing it because they encountered a problem and wanted to solve it. Everyone who has used a web browser and know how to code can contribute to the development of a new browser. But then there are *vertical* applications, which often demand specialised knowledge and the developers rarely use the

Table 1. Characterizing FOSS and OSS 2.0

Process	FOSS	OSS 2.0

Development Life Cycle	• Planning—"an itch worth scratching"
	• Analysis—part of conventional agreed-upon knowledge in software development
	• Design—firmly based on principles of modularity to accomplish separation of concerns
	• Implementation
	  ○ Code
	  ○ Review
	  ○ Pre-commit test
	  ○ Development release
	  ○ Parallel Debugging
	  ○ Production Release
	(often the planning, analysis, and design phases are done by one person/core group who serve as "a tail-light to follow" in the bazaar)
		• Planning—purposive strategies by major players trying to gain competitive advantage
		• Analysis and design—more complex in spread to vertical domains where business requirements not universally understood
		• Implementation subphases as with FOSS, but the overall development process becomes *less* bazaar-like
		• Increasingly, developers being paid to work on open source

Product Domains	• Horizontal infrastructure (operating systems, utilities, compilers, DBMS, web and print servers)	• More visible IS applications in vertical domains

Primary Business Strategies	• Value-added service-enabling
	• Loss-leader/market-creating
		• Value-added service enabling
		  ○ Bootstrapping
		• Market-creating
		  ○ Loss-leader
		  ○ Dual product/licensing
		  ○ Cost reduction
		  ○ Accessorizing
		• Leveraging community development
		• Leveraging the open source brand

Product Support	• Fairly haphazard—much reliance on e-mail lists/bulletin boards, or on support provided by specialized software firms	• Customers willing to pay for a professional, whole-product approach
Licensing	• GPL, LGPL, Artistic License, BSD, and emergence of commercially oriented MPL • Viral term used in relation to licenses	• Plethora of licenses (85 to date validated by OSI or FSF) • Reciprocal term used in relation to licenses

Figure 3.1: FOSS vs OSS 2.0 (Fitzgerald, 2006)

software they create (Fitzgerald, 2006). The healthcare sector is an example of a domain where such vertical applications are used. A software developer rarely have an interest in, or the skill required to, using software in order to analyse blood samples, while the biotechnician doing the analyses rarely have the skills to develop or customise the software he is using to perform the analysis.

Since proprietary software is backed by a company, or individual, and their commercial interest, as long as there is competition in the field and users willing to pay for the software or connected services, one should expect the software to be maintained and regularly updated. Without competition, there is always the risk of the software stagnating, such as what happened with the web browser market. After Microsoft gained a near complete monopoly on web browsers by bundling Internet Explorer in Windows, they stopped developing the browser and the web as a platform stagnated since there were no competing browsers that threatened Microsofts dominance, and they therefore had no interest in further developing Internet Explorer to support the evolving web technologies. It was not until the open source Firefox browser was released by the Mozilla Foundation and started to become immensely popular that Microsoft was forced to focus on improving Internet Explorer, and now the web browser market is highly competitive and innovative. But there are of course possible drawbacks to not having a commercial entity supporting an application; whereas a company will offer technical support to its customers, whether that support is free for buyers of the software or they offer paid support plans, a volunteer based FOSS product often relies on support provided by the *community* rather than the creators themselves.

3.2.1 Communities

> *"FOSS applications without strong community support will naturally struggle to reach the level of functionality provided by commercial applications" (Seebregts et al., 2010)*

One of the benefits of FOSS, is that if a company developing a FOSS product looses interest in maintaining and improving it, anyone interested can often just grab the code and continue the development themselves. But in order for this to happen, there needs to be an active community so the software can survive and thrive. While smaller programs and tools, such as *find* and *grep* in Unix and its clones, can survive for many years without active development, larger programs such as operating systems, web browsers or a health information system needs a community of active developers to add new features, identify and remove bugs and add support for new platforms, technologies and standards. Community building is therefore an important part of FOSS development, since both dedicated developers and a large user base is needed for a FOSS project to thrive. Many of the most popular FOSS projects, e.g. the many different GNU/Linux distros, has managed to build large dedicated and self sustaining communities without much of an active effort but others require a more active approach to community building. Even in the case of FOSS projects developed by commercial companies, support for the software is often left to the community[2]. But how does one go about building a strong and robust community? Communities often arise from other developers finding the project interesting, and then adds features to scratch their own itch or improve the quality of the existing code. With more features, and better quality, the software may become useful to more and more people, both your average computer user and other developers, and the more activity a project sees, the more developers tend to join since high profile projects are often more popular to work on (Lakhani and Wolf, 2005). Most FOSS projects maintain mailing lists and online discussion forums in order to enable interaction and communication between the community members, and these are usually great resources for anyone who uses the software and needs support with something related to it. The larger projects might also have regular conferences organised, whether by the project owner(s) or some dedicated part of their community. For instance, DebConf is a yearly conference for anyone interested in the Debian operating system, and allows users to meet with the developers and exchange ideas and experiences.[3]

The question then becomes that given that there exists both a commercial proprietary solu-

[2] E.g. the React Javascript Library developed by Facebook

[3] http://debconf.org/

tion, and a FOSS alternative that both fulfil a set of specified requirements, how does one decide between them?

3.3 Economics of FOSS

3.3.1 Software customisation

An argument that is often voiced in favour of FOSS, is that since the source code is freely available anyone with the necessary skills and interest can make modifications to the software so that it better fits their need, and make these changes available to everyone. Commercial software comes without this flexibility and adaptability, and, while some software products come with support for plugins, usually only provides new functionality through paid upgrades. For the consumer oriented market, this makes a lot of sense, since consumers often have overlapping needs and it is easier to just sell them a program that covers *most* use cases, but not all, since this helps keep the price down and the few users who needs *that* particular feature is not a big enough market to justify the extra investment needed. The enterprise market is different. When selling to large or medium sized businesses you might not just be selling software for a few hundred or thousands of dollars, an enterprise software system can easily cost tens or hundreds of thousands, even if it is a generic ”off-the-shelf” solution and not custom made for that particular customer. In this case, it makes a lot more sense for a software provider to accommodate a customers wishes for extra functionality which is not part of the core product. If the choice is between signing a multimillion dollar contract for a product, which development costs have already been covered, and spending a few hundred thousand on developing some new functionality on the one hand, and *not* signing the contract, because the customer is not interested since that particular functionality is lacking, on the other, it would most likely be an easy decision.

Demands from users for additional functionality cannot always be ignored, if the demand is high enough and there is a competitive market, other suppliers would seize the opportunity and provide software that satisfy the demand, so customer feedback will to some degree be taken into account during further development of the software. A study (Pollock et al., 2007) found that one software supplier segmented their user base into *Strategic*, *Consultative* and *Transactional* customers where the Strategic and Consultative customers were central to further development of the software, while Transactional customers were more peripheral. To quote the study, *”Transactional customers don’t want to spend money. They want everything for nothing. So for every day you put into them, you get nothing back. So you put your days into Consultative customers who want to work with and spend with you. Wheras Strategic are all about people who help share the vision of where the product is going to go over the coming years.”* Without having done any research on the subject, and therefore basing this purely on my own experiences, I would argue that this same distinction can be made when it comes to FOSS. Normal users of FOSS are the transactional customers, they complain in online forums and in conversations with friends that Firefox is too slow and uses too much memory, but they don’t bother submitting a bug report since that would require work on their part. Consultative users would be beta testers and power users, who contribute time and effort towards making the software better, by filing bug reports or pushing the software to its limits in order to identify its weak points. The Strategic users are users who submit their ideas, designs or code to the project and thereby contributing to the development and evolution of the software. If you want a bug removed from, or a new feature added to, a software product the worst thing you can do is complain about it without doing anything yourself, the next best thing is to provide detailed information about the bug, and the best thing would of course be to develop the requested feature yourself and submit it to the development team.

The problem remains however, both in the business software world and FOSS world, that even if you do your best at trying to be a valued Strategic of Consultative customer or user, the supplier might still just view you as a Transactional one. You must not only provide your provider with something, they must also consider your contribution to be beneficial to themselves.

3.3.2 Vendor lock-in vs open standards

"To guarantee free access by citizens to public information, it is indispensable that the encoding and processing of data not be tied to any single provider... the usability and maintenance of software should not depend on the goodwill of suppliers or on conditions imposed by them in a monopoly market" (May, 2006)

When deciding between using FOSS or proprietary software, one should be aware of the concept of vendor lock-in, since many software companies will entice users with easy to use products while never mentioning the difficulty of migrating to another system in the future if you should find yourself unhappy with the system somewhere down the line. Often the proprietary applications store their data in equally proprietary binary formats which cannot always be read by other, competing applications. For instance, the office suite Open Office was only able to read Microsoft Word documents because someone had gone through the process of reverse engineering the format and made his work available to others. Today, the format has the status of a standard and its specification is freely available[4] but it is extremely complex and currently Microsoft Office is the only office suite that supports it 100%.

Since FOSS usually have no interest in keeping their users from migrating to other systems, they tend to support a wide variety of data formats and standards to allow easy interoperability with other programs. This means that it is often a simple task to have several different systems running and sharing a common dataset between them. This "increases the range of functionality and applications of the combined whole" and allows one to choose the software best suited for a task without having to worry if data can be exchanged between the different systems.

This is also touched upon in the discussion about failures and problems facing ICT4D, with the existence of so-called "data silos"; information systems that collect data but does not make it available to other systems and/or projects. There are valid reasons for commercial entities in search of profit to keep their data to themselves, but it is more difficult

[4] http://www.ecma-international.org/publications/standards/Ecma-376.htm

to justify why NGOs and development agencies would want to hinder others in using data gathered during an ICT4D project.

3.3.3 Total cost of ownership

As has been mentioned, one great benefit of FOSS is that it is usually gratis to obtain and use. There are no license cost and you can download and make as many copies of the software as you please. Depending on the number of software installations required, this can be of great importance. The total cost of ownership, TCO, is the sum of all direct and indirect costs of a product or system over time[5]. It includes the purchasing price or licence fee of the software, as well as support plans, training of users, loss of productivity due to down time and so forth. TCO is important to consider when deciding on which product one should go for. While a license fee of a few hundred dollars for a copy of MS Windows and MS Office might not seem that much for a private individual or a business owner in the North, it might be prohibitively expensive for someone in the South. On the other hand, technical support, maintenance and training of users might be very costly in the North due to high labour cost and therefore contribute greatly to the TCO, while labour in the South is a lot cheaper and will perhaps contribute less to the TCO than the original purchase price of the software. A comparison of software licence fees relative to GDP per capita in 176 countries can be found in Ghosh (2003), where the author uses a simple calculation to estimate the labour cost of a single MS Windows and MS Office license. As an example, if a license of MS Windows XP and MS Office XP costs USD 560 it is equivalent to 0.18 months of GDP per capita in Norway, meaning a Norwegian would have to work for four or five days to afford a licence when the GDP per capita in Norway is USD 36,815. In order to afford that same license, a person in Tanzania, where the GDP per capita is USD 271, would have to work for 24.78 months. A single license fee can in this way be seen as equivalent to two years of labour cost. Choosing a FOSS alternative such as Libre Office would cost nothing up front, and the money saved could, depending on the number of copies of the software required, pay for eventual support, hire additional staff etc. This example is of course very simplified, the labour cost for knowledge workers in Tanzania is higher than the GDP per capita, which is only an average for the entire country, but it should serve to illustrate the general concept that the purchase price of any software or information systems is only one factor that has to be considered. And it works both ways, FOSS may be cheap to aquire, but if the software is not mature or turns out to lack vital functionality, the costs incurred by support and eventual customisation may far outweigh the purchase price of a commercial alternative. Also, one has to consider software piracy whereby copies of software such as Microsoft Windows and Office can be obtained illegally for free or a very low price. Because of piracy and the often limited contribution purchase price makes towards TCO, Heeks states that there is no clear evidence of cost savings if using FOSS (Heeks, 2005). But I have already argued that development should not be viewed purely in economical terms, so the question of whether FOSS makes sense economically is just one part of the equation. In the next section I will briefly look at some of the other contributions towards development that FOSS can make.

[5] https://www.gartner.com/it-glossary/total-cost-of-ownership-tco

3.4 FOSS and development

"A FOSS strategy may be the only viable policy that allows developing countries to maintain a measure of technological independence in ICTs" (May, 2006)

Previously, I introduced the concept of the Learning Economy which argues that DCs should be given the opportunity to learn for themselves how to do things and the opportunity to build upon what they learn so that they may develop their own set of skill and expertise with which to compete with the currently developed countries on the global market. FOSS allows you to take something that others built, learn its inner workings and improve or extend upon it to build something new. There are numerous examples of businesses and products being built with the help of FOSS, even some of the worlds largest technology companies; Google uses Linux to power most of their servers and Facebook is built using the PHP programming language which is FOSS-licensed. You can certainly build a product or company on top of proprietary software as well, but FOSS often offers a lower barrier of entry which can make all the difference when starting out since you do not have to worry about purchasing extra licenses or upgrades or limitations to what you can use the software for.

Even with piracy existing, and the software purchase price contributing relatively little to TCO, there is still an enormous cost involved, with numbers from 2004 showing that countries in sub-Saharan Africa purchased proprietary software totalling about USD 24 billion per year, mainly from US-based companies(May, 2006), so DCs that receive billions of dollars per year in development aid, have to spend billions buying software from the same countries they receive aid from.

I will conclude my discussion about FOSS here, and say that it may not be a magic bullet, but has long been a viable alternative when it comes to personal computing and is increasingly becoming an alternative in other domains.

Another phenomenon that can help DCs cross the digital divide, or at least narrow it, is the rapid spread and adoption of mobile devices, particulary the mobile phone. The more advanced mobile phones are basically small general purpose computers in a pocket format, and even some of the cheapest mobile phones, whose pricepoint make them within reach of a large part of the worlds population have features and capabilities that can make them a very valuable tool. In the next chapter I will take a brief look at some common software development methodologies, which will become relevant for my discussion later on and after that I will take a closer look at mobile technologies and the opportunities they can offer.

Chapter 4

Software Development Methodologies

Software development, or software *engineering*, is a field which is now more than half a century old, but is still not fully matured. Hence, there is some debate over whether one can actually talk about software engineering and construction with regards to the process of building a software system since using the term engineering leads one to compare the field with more traditional engineering fields. An engineer tasked with building a bridge, has a very clearly defined set of functional requirements and an equally well defined and thorougly tested selection of tools and materials with which to construct the bridge. In order to construct the bridge, the work has to be planned in detail beforehand, with little or no room for changing requirements once the work has started. Software engineers rarely have the same luxuries. It lies in the very nature of software that it is malleable and adaptable to changing circumstances and the tools and construction materials (programming languages, frameworks, operating systems) are constantly changing and evolving. As a result of this, it is difficult to arrive at any consesus amongst software developers for how best to go about creating a piece of software. There are no ”building regulations” so to speak, only a collection of ”best practises” and a set of developement methodologies to choose from.

Two well known approaches to software development, which are almost polar opposites, are the Waterfall model and what is called Agile development. These two methodologies should be familiar to anyone with experience from the software world, but I will give an introduction to both of them here as well as take brief look at the Spiral model. These methodologies are not only applicable to the *development* of software, but also to the application of software in the construction and implementation of information systems.

4.1 The Waterfall Model

The Waterfall model is based on traditional engineering approaches from the manufacturing and construction industries, where changes to a product at a late stage in its development incurs high cost. It is therefore considered beneficial to work out a detailed set of requirements and specifications for the product in question before any work is started on designing and creating it. It is one of the earliest, if not *the* earliest models for software development. If following the Waterfall model, there are a number of phases the development work follows in order, each one completed before moving on to the next. This is what gives the model its name, the idea that work should flow from top to bottom like the water in a waterfall. Several modifications have been made to the Waterfall model through the years, but the unmodified version has five distinct phases:

- Requirements specification
- Design
- Implementation
- Verification
- Maintenance

The idea behind this approach is that it is easier and cheaper to fix problems at an early stage of development than later on in the process. If all the requirements are specified before any design or coding starts, it is possible to discover and possibly fix conflicting or opposing requirements while they are still only an item on a list, instead of discovering it only when half way through the implementation process. It also places a lot of emphasis on documentation, thereby allowing new developers or designers on the team to become familiar with the project just by reading the documentation and reducing, or preventing, loss of knowledge if a project member leaves before completion.

While this approach to software development has its benefits, and it is regarded as suited for projects where the requirements and scope are fixed and the technologies well understodd, it is often seen as too rigid and unsuitable for many software projects.

4.2 Agile Development

The Agile approach is of a different nature. Instead of spending lots of time early on in the process in order to avoid change down the road, Agile embraces change and emphasis is placed on flexibility and adaptability in order to accommodate the changes that will inevitably occur. That is not to say that Agile does not include any kind of planning, no matter which approach is taken one needs to have some overarching idea and design to follow, otherwise it would just be chaos. To quote parts of the *Manifesto for Agile Software Development*[1]:

[1] http://www.agilemanifesto.org/

Customer collaboration over Contract negotiation

Responding to change over Following a plan

That is, while there is value in the items on the right, we value the items on the left more.

If following the principles of Agile development, one would focus more on spending less time specifying requirements and negotiating contracts before starting development, and instead strive for frequent meetings and discussions with the customer during all stages of development. One key idea behind Agile is that the customer or client does not always know beforehand what their exact requirements are. That therefore there will always be changes to the requirements and specifications during the development stage, so through an agile approach where nothing is ”set in stone” these changes are more easy to accomodate, and at a a lower cost, than would be the case if following the Waterfall model.

4.3 The Spiral Model

A third approach worth mentioning is the Spiral model, which does away with the step wise approach of the waterfall model but still views software developement as following a set of specific steps although in this model they are conceptualised as a spiral, where each completed sequence of steps feeds into the next. It takes a risk based approach, where a projects risks are evaluated and on the result of that evaluation determines the next steps. It can accommodate the Waterfall model as a special case, as well as other approaches such as evolutionary prototyping (Boehm, 1988). Using this approach, one constantly evaluates the remaining risks in the project and focus is placed on development that will remove the remaining risks thereby getting a more flexible approach than the Waterfall model, and placing more emphasis on dealing with the risks involved than functioning prototypes that is often the focus in Agile.

Choosing a development methodology is important in most projects, and can greatly influence the quality of the work. Each project undertaken should be evaluated and based on its perceived risks and challenges a proper method should be chosen.

After this small digression, we will now take a look at the different mobile technologies and what they can offer.

Chapter 5

Mobile Technologies

In this chapter I will present a selection of mobile technologies, mainly in the form of mobile phones and the communication networks they depend on. I will start by giving a general overview of the evolution and history of mobile devices and the different network technologies from the mid-nineties when mobile phones became mainstream and until today. The capabilities of the different types of mobile devices will be discussed and we will have a look at the trade-off one has to make when choosing one specific technology over another. We will look at the rising interest in health related applications of mobile technology an area collectively known as mHealth and other areas where mobile devices can contribute to development.

5.1 The case for mobile

For people who have never had access to a landline telephone, and therefore could only communicate efficiently with people in their immediate vicinity, the mobile phone is a revolutionary device since it enables real-time communication over vast distances, both in the form of text messages and voice calls. Compared to this, the Internet can be seen as more of an evolutionary step since it ”only” improves existing communication capabilities and for the average person, being the social animal humans are, a voice conversation with a distant loved one is probably far more important than the ability to search on Google and read Wikipedia. But of course, mobile phones are getting ever more advanced and many now have the capability to access the Internet as well. We in the North are a bit spoiled when it comes to ICT, as almost everyone has, or can easily gain, access to a personal computer so when we talk about giving people access to computers and the Internet we tend to think of computers as laptops and workstations, equipment that is expensive and power hungry. But the truth is, that an incredibly large percentage of the worlds population actually do have access to a computer and the Internet, through their own mobile phone.

The last decade has seen an incredible adoption rate of cell phones around the world, as illustrated by figure 5.1. A report from the UN estimated the number of mobile subscriptions world wide would reach 6.8 billion by the end of 2013 (United Nations, 2013, p 56). The GSMA Intelligence, a part of the GSM Association which represents the interests of mobile operators worldwide, reports that 7 billion subscriptions were reached in April 2014 and they "[by years end] expect global connections to match the 7.2 billion global population total projected by the United Nations." (GSMA Intelligence, 2014). Not only that, but the mobile penetration rate in developing countries was expected to reach 89 percent by the end of 2013, and developing countries now account for over 77 percent of all the worlds mobile-cellular subscriptions. Considering that a survey across 24 countries reported that a median of only 23 percent of the population said they had a working landline telephone in their house, these numbers are truly astounding. In Ghana and Kenya, as few as 1 percent reported having a working landline phone while 79 and 82 percent owned a cell phone (Pew Research Center, 2014).

When discussing numbers such as these, it is important to note that there is not a one-to-one relationship between individuals and subscriptions. There are few individuals who would need more than one landline subscription, but having multiple mobile subscriptions is quite common. Also, a mobile subscription can be shared amongst multiple individuals, just like a landline, so the numbers just mentioned are not a representation of how many people who definitely have access to a phone, but can be used as a basis for estimates.

5.2 Overview of mobile technologies

When reading about mobile devices and communication networks, one encounters a lot of different terms and abbreviations that are often used interchangeably so it can be useful to define some of the most widely used ones. First off, there is the mobile devices themselves. Since a mobile device is, by the definition of the word, any device that can easily be moved around and carried by a person I will narrow it down a little and define a mobile device as a small, hand held computing and/or communication device. This definition rules out laptop computers which, although definitely mobile, have more in common with a desktop computer than a mobile phone and is not meant for handheld use. This rough definition leaves us with devices such as mobile phones, tablets and PDAs.

The mobile phone, or cell phone as the Americans know it as, is the device I will focus on the most and the one I refer to when using the term "mobiles". They can be divided into three main categories; *dumb phones*, *feature phones* and *smart phones*, each of which will be defined in the next section.

When talking about mobile internet access, I mean internet access provided by the mobile cellular network, as opposed to internet access via Wi-Fi.

5.2.1 Devices

Mobile devices come in many shapes and sizes and some are short-lived while others have stuck around in one form or another for decades. The most widely used mobile device is the mobile phone, which in its early days were big, clunky and expensive and mainly used by businesspeople and others who needed to be available at all times and could afford them. Ever smaller and cheaper devices, as well as better mobile cellular network coverage and

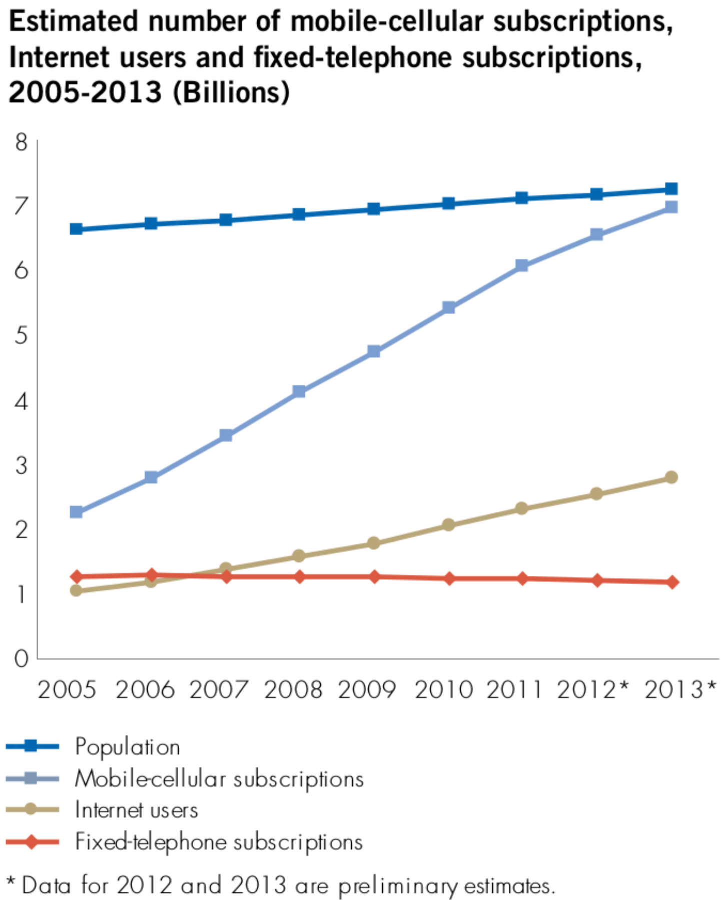


Figure 5.1: Global mobile subscriptions [Millenium Development Goals Report 2013 p.56]

cheaper calls, made mobiles affordable, and useful, for the large majority of the population in the global North in the late 90s and early 2000s. Parents saw the benefits and peace of mind it gave them to let their children carry a mobile and it wasn't long before teenagers became the dominant users of mobiles. These early mobile phones, which main function was to make calls or send text messages, is sometimes referred to as *dumb phones*, meaning that they had little or no capabilities outside providing voice communication.

For many years, dedicated professional devices called Personal Digital Assistants, or PDA for short, existed alongside the mobile phone. These PDAs provided functionality that the then current mobiles could not, such as email, internet access via Wi-Fi and touch input via stylus. Gradually, many of these features made their way into mobile phones and the mid 2000s saw a lot of experimentation in form factors and functionality in mobiles; everything from dedicated gaming mobiles to camera phones and email-focused phones such as the Blackberry. Common to almost every mobile in this time is the extra functionality and capabilities they introduced. On-board cameras and media players became common, as well as colour screens, basic web browsers, user installable apps and GPS. We can refer to these phones as *feature phones*, meaning that they have extra features and functionality that extends their use beyond that of merely a mobile telephone.

The design and functionality of mobiles went in a completely new direction when Apple introduced the iPhone in 2007 and popularised the large touch-screen design. Ever since then, mobile phones have become more of a powerful all-purpose pocket computer which coincidentally is also able to make phone calls. They make up the third category, the *smart phones*.

A relative new-comer amongst mobile devices is the tablet computer, or just tablet, which can be viewed either as a large mobile phone, without the telephone functionality, or as a laptop without the keyboard and broad selection of ports. While tablets have existed in one form or another for many years now, they were a niche device for people with special needs before Apple popularised them with the introduction of the iPad in 2010. Until then, tablets had mainly been ”normal” computers in a tablet form factor, running a regular computer operating system like Microsoft Windows and having little to no software written specifically for the tablet form factor. This made the early tablets somewhat cumbersome to use since the software had a user interface designed for mouse and keyboard operation, while the tablet had a touch screen as its main input interface. The iPad did not try to be a normal, full blooded computer, and ran the same operating system as the iPhone, an OS designed for mobile use and touch screen based input. This approach, a big screen with an intuitive user friendly interface and lots of apps, made the iPad immensely popular. Although public interest in the tablet form factor seems to have been reduced lately, and tablets are not selling as well as they used to, there is still many different tablets to choose from and they can be a great tool in many circumstances.

5.2.2 Networks

Since a mobile phone's primary purpose is communication, it is important to have access to a mobile cellular network if you want to make full use of it. Mobile networks have a long history, and there are many different standards, but the most widely spread and well known standard is probably GSM, the Global System for Mobile Communications. Developed by the European Telecommunication Standards Institute it was intended to replace the early analog mobile networks with a more modern digital one. Being a digital network rather than an analog one, allowed GSM to support data transfer in addition to voice calls. This led to the development of the Short Messaging Service, SMS, which allowed users to send short text messages over the GSM network. SMS proved immensely popular amongst users and have been a major source of revenue for mobile telecommunication providers for many years. GSM later expanded the data transfer capabilities with the introduction of packet data transfer via GPRS and EDGE and thus allowing mobile phones to connect to the internet through the cellular network. GSM is often labeled “2G” since it was a second generation mobile cellular network, and its analogue predecessors is labeled 1G. Following in that vein, GPRS and EDGE were labeled 2.5G and 2.75G. In 2001 the first third generation networks appeared, based on the UMTS standard they are usually just called 3G networks.

Generation	Name	Capabilities	Data rates
1G		Voice calls	None, analog network
2G, 2.5G, 2.75G	GSM, GPRS, EDGE	Voice&video calls, SMS, MMS, basic data traffic	Few Kb/s to hundreds
3G	UMTS	Voice&video calls, SMS, MMS, high definition video	Hundreds Kb/s to several Mb/s
4G	LTE	Voice&video calls, SMS, MMS, several HD-video streams, large data transfers	Several Mb/s to hundreds Mb/s
5G			Gb/s?

Table 5.1: Mobile networks and their capabilities

Table 5.1 gives and overview of the different standards and their data rates and capabilities. Within each generation there are several other standards and abbreviations, too numerous to list all of them here so I have only included the most well-known. The different generations of mobile networks offer different services. The oldest, GSM, offers voice calls and basic data transfer in the form of SMS and MMS. Data could either be transmitted as plain text or compressed SMS, but there was a limit to how much data one could reasonably transmit since an SMS is limited to 160 ASCII characters and the network operators usually charge a fee per SMS. Starting with GPRS and EDGE it became possible to transfer arbitrary data over the mobile cellular network, albeit with very slow data transfer rates. Dedicated applications could now generate and capture data on the device and transfer it over the GPRS connection in a compressed form and it allowed devices to browse the web with basic web browsers. The introduction of the 3G network brought with it high speed data transmission and 3G enabled devices such as mobiles or dedicated modems could rival dedicated internet landline connections. Currently, 4G is the most advanced mobile network standard in regular use and while it brings with it higher data transfer speeds than 3G it does not really add anything new.

5.2.3 Operating Systems

Just as there were almost as many different computer operating systems as computer manufacturers in the early days of personal computing, there was a multitude of different mobile operating systems until quite recently. Nearly every manufacturer had one or more mobile operating systems of their own so the functionality and capabilities, as well as the user interface, of mobile phones were different depending on the manufacturer. There were even large differences between models from the same brand. Software often had to be written targeting a specific device, and users rarely had the option to install apps of their own. While there is still a large number of operating systems for dumb and feature phones, smart phones are a different story. If you want to buy a smart phone today, you basically have three operating systems to choose between: iOS from Apple, Android from

Google, and Windows Phone from Microsoft. While iOS is limited to Apple products only, which are expensive premium devices, and Windows Phone to products from Microsoft and Nokia, Android can be found on devices from a range of different producers, and both WP and Android phones come in a wide price range. All three operating systems have online app stores where you can download applications for your device, and tie in with different software and services from the operating system provider to form a so called eco-system around the platform. The amount of user installable apps, and usefulness of the services provided, means that you can easily find yourself locked in to one eco-system and find it difficult to switch to a different platform later on.

5.2.4 Different Solutions

If one wants to use mobile phones to gather and report information, the different devices and networks offer different capabilities for transmission of data. Since all network generations and mobile phones support voice calls and SMS, the applications with the broadest reach would be the ones that rely on these technologies. Applications can transmit data via SMS, either via an application installed on the users phone which generates SMS reports, or with the user manually composing an SMS based on some pre-defined structure. There is also the possibility for interactive voice calls, where the user dials a number and is presented with spoken instructions. Everyone who has ever called a customer support center should be familiar with these kinds of systems; where a caller is given several options and selects one by entering a number on the keypad. Both of these approaches work, and are in use, but common to them both is that they severely limit the amount of data that can be sent, and the type of data. They can also be very slow and cumbersome to use, if the user has to compose an SMS for every report or go through a long list of options in an interactive voice call. Feature phones provides some better options, by allowing the user to install applications, usually based on a Java platform for embedded systems, the Java 2 Platform, Micro Edition (J2ME). These apps can present users with pre-defined forms which they can fill in, thereby removing the need for typing in long SMS's with difficult to remember encodings. Such phones also allow storage and compression of data on the onboard memory so that it can be transferred at a later point in time if network coverage is not immediately available, making them very useful in areas with poor coverage or in the event of a temporary network outage. The most capable mobile devices, are the smart phones which are equipped with powerful processors, high resolution cameras and other features that make them very useful devices if utilised correctly. They are equipped with advanced web browsers that allow off-line storage of data, which means that applications do not necessarily have to be installed on the phone itself, but can be web-based and accessible through any device with a compatible web browser. The benefit this provides is that it removes the need to update the application on every device that uses it, one only has to update the application running on the web server and then new functionality or bug-fixes is instantly available to everyone accessing it.

5.3 Mobile Technology in Developing Countries

The use of mobile phones have many clear advantages; most people already own one, they know how to use it, it has several different uses so it does not require the introduction of a new device used just for a single purpose. [Compare this to telecenters and efforts to introduce regular computers to rural populations etc. Mobiles let you call and text friends and family, what use is a computer when no one you know have one and you have no need to compute something?] . Making it possible to record and submit data using mobile phones offers many advantages in rural, and even urban, areas in DCs. More traditional computer equipment, such as desktop and laptop computers are more expensive, power hungry and in need of an Internet connection. Mobile phones on the other hand are cheap, can last days on a single charge, can be charged easily with everything from solar cells to car batteries, and they are very robust since they are designed to be carried around in your pocket for days on end. As was discussed in the previous chapter, some reasons for ICT project failure in both DCs, is exactly the lack of stable electrical supply, non existent or intermittent Internet connectivity and the lack of spare parts when equipment breaks down. Whereas both laptop and desktop computers have moving parts that can suffer mechanical failure, like hard drives and fans for cooling, and therefore often need to have parts replaced from time to time, mobile phones do not The low power requirements of mobile phones, many have standby times of over a hundred hours or more means that they are less affected by a power cut and their reliance on the mobile cellular network for connectivity rather than Wi-Fi ensure that they are still able to function as long as there is still a working cellular tower nearby. Most smart phones, and more advanced feature phones, are even able to share their data connection by creating an ad hoc Wi-Fi network that other devices can connect to, making it possible to continue working online during a power outage, as long as the batteries last.

5.4 Trade-offs

When choosing a specific technology on which to base a mobile application, it is important to consider the strengths and weaknesses of the selected technology for they each have their different ones. Trade-offs usually have to be done between four important properties: robustness, flexibility, financial cost and usability. In figure 5.2 we see a selection of different solution types positioned in a two-dimensional solution space constructed by the dimensions flexibility and robustness. A solution's overall usability is represented by the size of the ellipses, while financial cost is not considered in this figure (Sanner et al., 2012). We see that there is no single solution that satisfies the dual demands of robustness and flexibility and that generally the more flexible a solution is, the less robust it is and vice versa. The solutions that arguably show the most promise is locally installed applications using mobile data transfer, and web based applications accessed through a mobile browser. However, custom applications and advanced mobile browsers require more expensive mobile phones than the solutions based on voice calls and SMS.

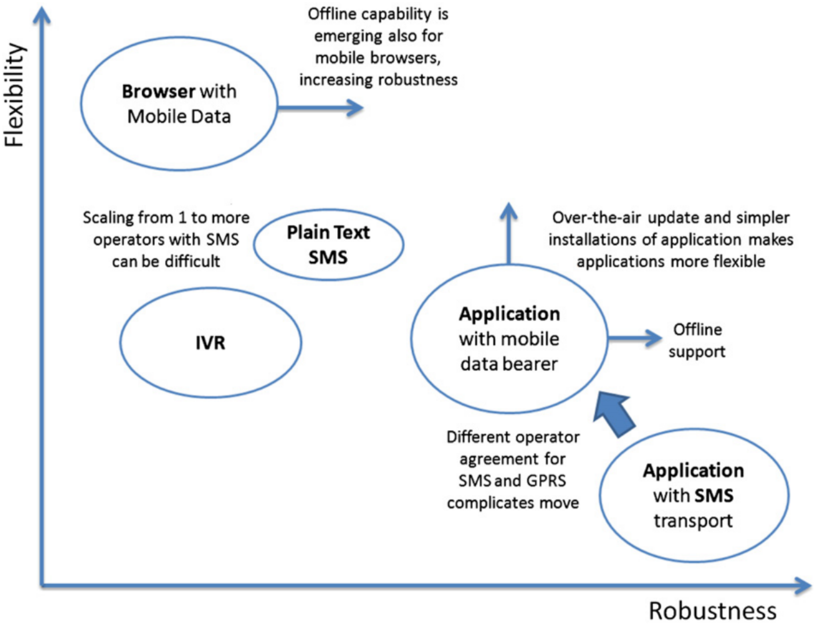


Figure 3 Trade-offs between solution type flexibility, robustness and usability.

Figure 5.2: Trade-offs between solutions (Sanner et al., 2012)

5.5 Using mobile technologies for development

Both industry and development agencies see a lot of potential for mobile technologies in developing countries. One famous example of how mobile technology can be applied to an existing problem is M-pesa, the mobile money system developed by Vodafone. M-Pesa gives access to banking services to those who have never had a traditional bank account, and may not be eligible to get one. It has spread rapidly across sub-Saharan Africa, and nine countries have more people with a mobile bank account than traditional bank accounts (Penicaud, Claire and Katakam, Arunjay, 2013). M-Pesa empowers those who previously only had access to cash money, and gives them opportunity to transfer funds quickly and over large distances, thereby increasing their economic freedom.

A report by UNICEF, examines how mobile technology can be used to increase the low literacy rate in many developing countries (West and Chew, 2014). Literacy is an important factor in achieving universal primary education, which is one of the Millenium Development Goals. The online encylopedia Wikipedia, is partnering with mobile carriers and operators to provide access to Wikipedia free of charges.[1]

But the area that has perhaps seen the most interest, is the application of mobile technologies in the health care sector.

[1] https://wikimediafoundation.org/wiki/Wikipedia_Zero

5.5.1 M-health

A rising interest in using mobile devices and technologies to provide better health care, can be seen both in developed and developing countries. A quick search on Google will yield numerous results in the form of news articles and business case studies related to mobile health applications, or as it is often called, m-Health. Not only do we see the development of health related smart phone accessories which is becoming very popular in West, e.g. FitBit, Apple Healthkit and various other devices, but there is also a lot of innovation happening in DCs where appliations are being developed not only for smart phones, but for feature phones as well. Based on the rising interest in personal health and the ”quantified self”, PWC is predicting a global mHealth market of USD 23 billion in 2017, up from USD 500 million in 2010. Most of this growth is expected to occur in the developed countries, while Africa will account for only 5% (PWC, GSMA, 2012).

While m-Health in the developed countries is mainly focused on the individual, with continuous monitoring of health indicators, utilising super computers to diagnose a patient and similar, the use of m-Health in DCs focuses more on providing the population at large with better health care. Personal devices like FitBit and similar might be of interest to those who can afford it, but for governments, NGOs and aid agencies there is more interest in using mobile devices for gathering and reporting health data and providing health related information to the public. A popular use case for mobile phones in DCs is SMS-based systems for disease prevention and control, where data about disease outbreaks is crowd sourced and allow authorities to detect possible outbreaks and monitor them if they should spread (Déglise et al., 2012b; Freifeld et al., 2010; Déglise et al., 2012a).

This marks the conclusion of the literature study that provides a motivation for this thesis. The next chapters will present the context around the specific case that I worked on, starting with an introduction to HISP and the DHIS2 software, before introducing Zambia and the organisations I worked with there and then move on to a case description and finish with a discussion about the work that I participated in.

Part II

Context, Case and Method

Chapter 6

HISP and DHIS2

This chapter will present the Health Information Systems Programme, HISP, an ongoing effort to improve HIS in developing countries. Central to the work HISP is doing, is the HIS they develop, the District Health Information System, which now in its second major version is called DHIS2. I will start by giving a brief introduction to the motivation and goal of HISP, before presenting a summary of the programme's history. Since HISP and the DHIS is intrinsically linked they will both be covered in this history section but will otherwise be treated seperately. An overview of the HISP organisation follows, along with the presentation of the theory of Networks of Action which HISP places much focus on. I will then look at the DHIS2 in particular, its purpose and functionality as well covering the development process and the technologies upon which it is based.

6.1 Introduction to HISP

HISP is a "global network established, managed and coordinated by the Department of Informatics at the University of Oslo."

"The overall goal of the Health Information Systems Programme (HISP) at the University of Oslo 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." (HISP UiO, 2014).

It was started as a research project, funded by the Norwegian government, in collaboration with the University of the Western Cape, UWC, in South Africa after the fall of apartheid in an attempt to improve the extremely fragmented health care system in SA. Today, more than 20 years after its inception, it has grown into a global network with [nodes] all over the world. As the HISP mission statement says, it aims to strengthen the health care systems in DCs world wide, while at the same time working to build local capacities so that the countries in question can continue the work on their own, or help other countries, without having to rely on external expertise and/or donors. The project focuses on three strategic focus areas:

- Strengthen HIS
- Develop and govern DHIS2
- Traning, education and research

6.2 History of HISP and DHIS2

HISP is the result of the Scandinavian union based participatory design approach meeting grass root level anti-apartheid politics in South Africa in 1994 (Braa and Sahay, 2012). After the fall of apartheid in SA, the new government saw a need for improving the health care system which during the apartheid years had been inequitable, fragmented according to race and heavily centralised. Work was begun on a HIS based on Microsoft Office 97 technology and written in Visual Basic, both because of powerful features for working with large datasets, such as pivot tables, and also because users and developers were already familiar with the technology. This first version of the DHIS was deployed in SA in 1996 and became a national standard in 2000. Working from the concept of "information for local action" the system was designed to be decentralised and have features of flexibility and local control deliberately incorporated in the software in order to empower local users.

From 2000 and onwards, scaling both geographically and across areas of focus such as software and education became the key efforts. Problems were encountered when HISP tried to introduce the same solutions that had worked so well in SA, in India, Cuba and Mozambique. Both in Cuba and India there was resistance against the concept of a decentralised HIS, likely attributable to the centralised nature of Soviet style governments in Cuba's case and the bureaucracy of British colonial rule in India's. These attempts made HISP realise that context matters and that the approaches to design and development of HIS must be flexible and broad based enough to adapt to different contexts.

When trying to answer the question of how to deal with the limits of scaling, both in regard to the technical systems and approaches to development and implementation, HISP points to the solution in terms of consciously trying to cultivate networks within which collaborative action can be shaped and nurtured in order to address the twin problems of scale and sustainability. This gave birth to the theory of networks of action, and if one is to conceptualise HISP as a network it is comprised of various entities such as Universities, Ministries of Health, international agencies like Norad and WHO and in-country implementing agencies such as HISP-SA and HISP-India. The network is never static, there is always new memberships and attempts to form partnerships with other OSS development groups such as OpenMRS and iHRIS. Local skill and competence was considered important for the longevity and sustainability of the project, and Universities were recruited to establish Master programmes and PhD education which became an integral part of HISP. Figure 6.1 serves as an illustration of this network.

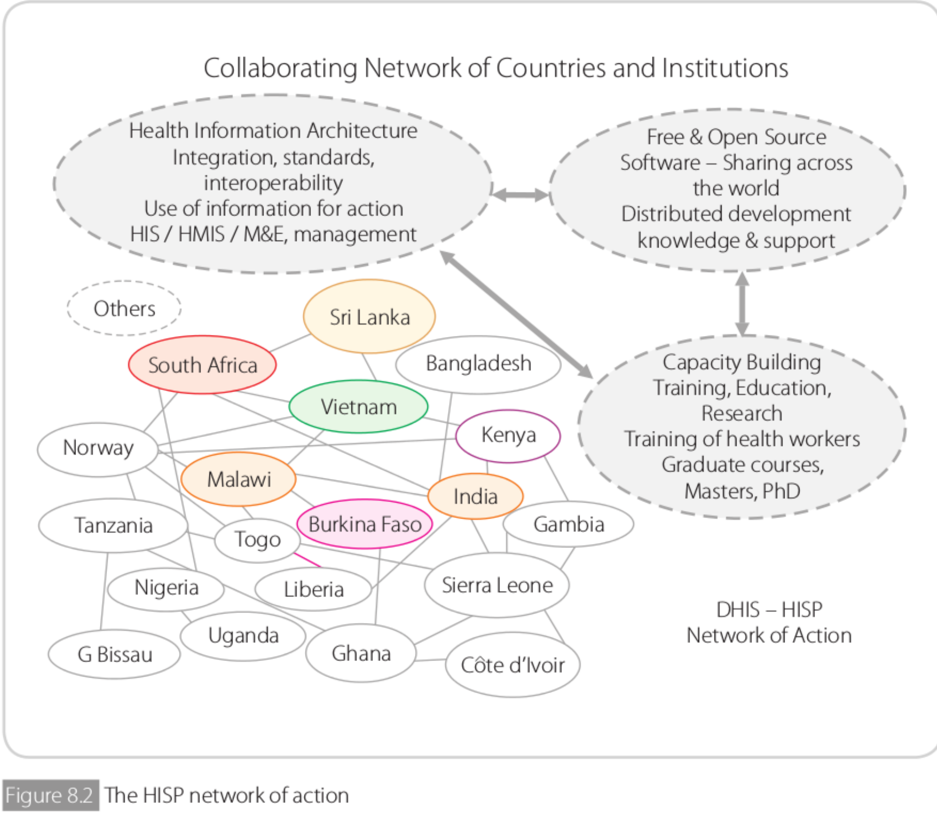


Figure 6.1: The HISP network of action (Braa and Sahay, 2012)

Geographical scaling of both HISP and the DHIS made evident the limits of DHIS in its current form. Rapid prototyping and the iterative design process had resulted in the a rigid and overall messy architecture. To address these problems, a completely revised and internationalised version of the software was developed in 2004, resulting in DHIS version 1.4. South Africa did not adopt the new version but kept on using version 1.3 since this was stable, established and well adapted to local needs.

After the introduction of this new version, the two-person development team in SA became a bottleneck in the effort to support an expanding network of users and requirements, and while the development took place in SA the users were now primarily elsewhere. This fragmentation of the software, along with the current architecture itself not being suitable for distributed development and the fact that no code sharing tools had been taken into use because of the small and co-located development team, meant that something had to be done. Additionally, dependence on the Microsoft platform was seen as problematic since it required the full MS Windows and MS Office stack to run, even though DHIS itself was open source. Redevelopment of the software was therefore started in 2005, this time under the leadership of UiO, and the aim was to distribute development activities amongst the countries participating in the HISP network in order to bring software development closer to the context of use. This resulted in DHIS2, a web based software platform built on "bleeding edge" Java technologies and modern web standards.

After the launch of DHIS2, HISP has seen a rapid increase in the adoption rate of the software which is now being used in about 50 different countries across dozens of projects run by both governments and NGOs. A timeline of notable events in HISP history is shown below.

- 1994 – HISP is started
- 1996 – First implementation of DHIS version 1 in South Africa
- 2000 – National standard in South Africa
- 2000 – HISP expands to India, Cuba and Mozambique
- 2004 – DHIS version 1.4 released
- 2005 – Development started on DHIS2
- 2006 – First instance of DHIS2 deployed, in Kerala India
- 2010 – First national online DHIS2 instance implemented in Kenya, Ghana and then Rwanda
- 2010 – HISP receives funding from Norad to employ full-time developers to work on DHIS2 so countries can be more confident of its longevity
- 2011 – First DHIS2 Academy is held
- 2012 – Joint 3-donor effort (PEPFAR, Global Fund, Norad) to strengthen DHIS2 use in developing countries
- 2015 – 47 countries using DHIS2, national standard in 16 countries

6.3 The HISP organisation

HISP UiO is the governing body and central node in the HISP network. The team at UiO manages both the overall HISP strategy, activities and interaction with external partners as well as leading the development of DHIS2. The software developers at UiO are supported by developers in Ireland, USA, South Africa, India and Vietnam. In addition to the software development teams, there are several expert nodes with very experienced DHIS2 implementers which have well-established communication with the core team in Oslo. The large geographical distances between the different HISP nodes requires that most communication and cooperation is handled via email and services such as Skype.

6.3.1 Financing, partners, students

HISP has from its inception been financed by the Norwegian Agency for Development Cooperation, Norad, and development of DHIS2 is currently sponsored by UiO, Norad, The Research Council of Norway, The Global Fund, and PEPFAR. Norad provides core funding to the HISP UiO "Centre of Expertise" and targets the core development of the platform and coordination and support of implementation in countries. The Global Fund is supporting implementation of DHIS2 in countries around the world, and PEPFAR has started using DHIS2 for internal reporting from their activities and provides funding to further development of DHIS2 to support their specific reporting requirements.

6.3.2 Capacity building

HISP describes itself as a capacity building project, in the sense that it does not just develop a complete HIS and help implementing it, but it also places a lot of effort into building local capacities. To this end, HISP has been working with local governments and universities in several countries to establish national Master and PhD programmes, which are now present in at least six countries. PhD students from participating countries spend their final year at UiO under supervision from the HISP leadership team. Over the last 10 years, 35 PhDs have been awarded within the HISP network and there are currently about 40 PhD students attached to the project world wide. In Malawi there is efforts underway to integrate DHIS2 training in undergraduate education, from nurses to doctors.

Building local capacities is central to the goal of HISP and education, research and training of users are seen as important activities. In order to train local users and enable participating countries to organise their own training and education of users, HISP held the first DHIS2 Academy in 2011. Aimed at training health workers and officials in the use of DHIS2, and also "training of trainers", these academies are now held several times a year as regional academies in different parts of the world. Early on, they were run with the help of several HISP-staff from UiO, but the first Academy run completely by local staff has already been held, and the hope is that they can in the future serve as a source of revenue for the network nodes that conducts the Academy.

6.4 DHIS2

6.4.1 The data warehouse

DHIS2 aims not only to be a fully functional HIS, but also to work as a data warehouse where data from many different sources can be stored and aggregated and work as a platform upon which to build additional applications and services, as illustrated by figure 6.2. DHIS2 does not support the recording of individual health records, but is instead concerned with collection and aggregation of different health data and indicators for the population at large and provides tools to analyse and display these.

A problem faced in many countries is that a lot of pilot projects are started, some of which survive and thrive while others fail, and they often collect and generate useful data but this data is never shared outside the organisation running the pilot project. This is

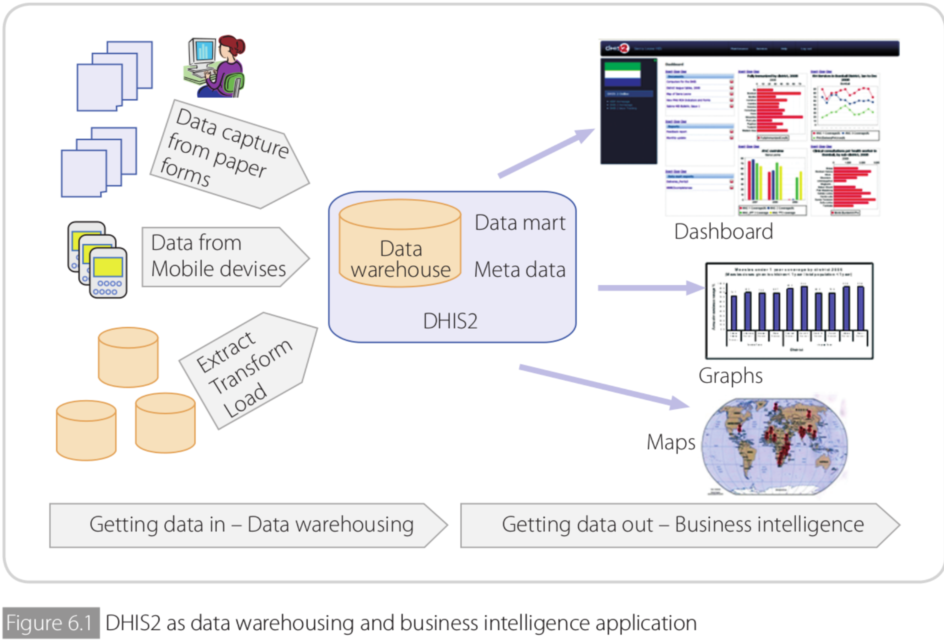


Figure 6.2: DHIS2 Data warehouse (Braa and Sahay, 2012)

not necessarily because the people responsible for the project does not want to share their data, but rather because there is no easy way to do so, so what we get is the concept of information silos where data is stored somewhere but perhaps never used or shared with others, and possibly in a format that is incompatible with others (Hoe, 2006, p 60-61).

6.4.2 Technology

DHIS2 is a web application built with standard Java technology, which means that it can run on any Java enabled web server or container and is accessible through any modern web browser over the Internet. An advantage to this web based approach is that DHIS 2 can be deployed on an online, national and central server with clients connecting to it via the web, as a stand alone application running on a single computer as well as a local server for an offline intranet setting. Since applications written in the Java programming language are platform independent, DHIS2 can run on any platform for which a Java Runtime Environment, JRE, is available. This allows DHIS2 to be deployed on practically any popular computer platforms, including Windows, Linux, Mac OS X and Solaris.

DHIS2 follows the W3C standard for HTML and CSS and can therefore run on any standard compliant web browser, but because the application makes heavy use of Javascript to display and render content, the development team recommends using the Google Chrome browser because of its Javascript performance.

When deploying DHIS2, one can choose one of several different database systems to run it on, currently it supports the PostgreSQL, MySQL and H2 database systems. And with only minor development efforts it can be made to run on any major relational database. Developers do not access the underlying database directly, but interact with it through the supplied Java API.

DHIS2 is distributed in several formats, which are all available for download from the projects web site. DHIS2 Live is a complete package containing everything needed to install and run it. The package contains the DHIS2 software, together with an embedded Jetty servlet container and an embedded H2 database. It is also possible to download a WAR file, which requires the installation of a Java server such as Tomcat or Jetty and a relational database. This version is recommended for server setups and environments with high traffic and data volumes. Those who want to configure their own installation can download the source code and build the application with the help of the Maven build tool which allows the removal or addition of modules via an XML-configuration file. Also available are a Java mobile client which can run on Java enabled feature phones. The mobile client relies on an available data connection such as GPRS, EDGE or 3G over which it communicates with an online DHIS2 server instance. A mobile app for Android based devices are available as well, and can be used for data capture and profile management.

6.4.3 Organisation of development

In its first few years, DHIS2 saw a lot of development done by master students attached to the project, but ever since 2010 the project has received funding for full time developers from Norad.

Development of DHIS2 is mainly done by three core developer teams, located in Oslo, Norway, in India and in Vietnam. The lead developer is located in Oslo and the team there is mainly focused on the core functionality of the software, while the development team in India is mostly working on implementing features they need for their own use. The source code is hosted on the online software collaboration platform Launchpad [https://launchpad.net/dhis2] which supplies the project with code hosting using the Bazaar version control system, bug tracking, mailing lists and other functionality. As is common with many FOSS projects, all source code, bug reports, blueprints for future development and mailing list discussions are available for anyone to read a the project homepage. Two mailing lists are maintained, one meant for developers and one meant for users, but both of which can be subscribed to by anyone interested and they both see a great deal of activity. Since DHIS2 is FOSS, it means that everyone who wants to can freely download and examine and edit the source code, and the permissive software license it is released under allows anyone to reuse any part of the code in projects of their own. However, if anyone should want to contribute code to the project they will have to submit it to one of the core developers, who are the only ones permitted to merge code into the main repository. New versions of DHIS2 are released every few months, and it has at the time of writing seen 21 releases since version 2.0 putting it at version 2.21 at the time of writing.

In addition to development work being done by full-time employees and master and PhD students attached to the project, DHIS2 has been a Google Summer of Code project for several years, and it is moving away from being a single monolithic application with a specific use case in mind, to being more of a platform which can be adapted and built upon to suit many different needs.

Chapter 7

Zambia

This chapter will give a short introduction to Zambia as a country and how the country and its healthcare system is organised before presenting some background information on Akros and ZCAHRD.

7.1 About Zambia

The Republic of Zambia is a landlocked country in Southern Africa and is perhaps most known for the mighty Zambezi river and the Victoria Falls, which it shares with its neighbour Zimbabwe. The country gained its independence from the United Kingdom in 1964, and with a large proportion of its exports being copper experienced a significant drop in income and a descent into poverty after international copper prices declined in the 1970s. Recent years have seen it become one of the worlds fastest growing economies, but it is still one of the worlds least developed countries with infant mortality rates amongst the highest in the world and life expectancy one of the lowest. The population is very urbanised, with over 40% living in cities. As many other African countries, the Zambian population is composed of a multitude of ethnic groups and a total of over 70 languages and dialects. The official language however is English, which is used in schools and to conduct government business.

The country, a map of which is shown in figure 7.1, is divided into ten provinces, which is further divided into a total of about 90 districts with Lusaka as the capital city and its center of government (Zambia Central Statistical Office, 2014).

Zambia is still a very traditional society. Besides the official government there is still the old tribal system with chiefs and local headmen. In rural areas, outside the cities and major population centres, the official government does not always have a large presence so much of the administration of those areas are left to the local chiefs and the local population often listen more to the words of their chief than the national government. This means that any development project aimed towards these areas need not only approval from the Zambian government, but also the acceptance and cooperation of the local chiefs and their

Figure 7.1: Zambia: Provinces and Districts (Zambia Central Statistical Office, 2014)

headmen.

A maintained effort to fight against malaria has led to the successful eradication of malaria in several districts in Zambia, and the few cases that occur in these districts are often found to stem from people travelling to that district from a more malaria-prone area. Lusaka and Livingstone, the countries major tourist hub, have low rates of malaria infection while the Copperbelt region, bordering the Democratic Republic of Congo to the north, still struggles with high infection rates.

Zambia Healthcare System

Health services in Zambia are provided by government institutions, church institutions, mining companies and other companies. In some parts of Zambian society, traditional medicine is still quite popular, and even in Lusaka one can see signs along the roads advertising the services of traditional healers. The treatments offered can range from spiritual healing, based both on the Christian faith and traditional belief systems, to the use of herbs and other traditional medicinal substances. Apparently, there is little to no conflict between the traditional healers and modern medicine. People in rural areas will often turn to traditional medicine if the nearest clinic is too far away, or use it as an alternative, or supplement, to modern medicine if the treatment they are given is not working properly. But most will seek out modern health care where and when it is available.

The government health care system is comprised of five levels (from top to bottom)[1]:

[1] http://www.moh.gov.zm/docs/facilities.pdf

- Third level hospitals. They cater for a catchment of >800,000 people. In 2012 there were six of these in the country.
- Second level hospitals. There are 19 of these, and they are found at provincial level and cater for between 200,000 to 800,000 people.
- First level hospitals. A total of 84 of these, which are found in the districts. Cater for 20,000 to 80,000 people.
- Health Centre (HC). There are two types of health centers: urban health centers, serving 30,000 to 50,000 people, and rural health centers serving about 10,000 people. There are 409 urban health centers and 1,131 rural health centers.
- Health Post (HP). Built in communities far away from health centers. There are 307 of them, each of which cater for a population of about 3,500 in rural areas and 1,000 to 7,000 in urban areas. They offer basic first aid rather than curative health services.

In addition to these official services, there are also thousands of volunteer community health workers (CHWs) working in rural areas. They provide basic health care at village level, and keep small stocks of medicine and necessary equipment. People can seek them out at their home, and they also actively seek out patients suffering from illness and disease. Zambia aims to have each CHW cater to about 500 people, but currently they do not have enough CHWs to reach that goal. More recently, Zambia has created a new group of health workers they call Community Health Assistants (CHAs). These workers receive more thorough training than the CHWs and will assist health post staff and CHWs in their work. Full national CHA coverage is expected to take several years.

7.2 Akros

Akros is a small NGO founded by an American couple and based in Lusaka, Zambia. Their stated mission is to "...[strengthen] national health systems in developing countries" and in Zambia they work mainly with malaria prevention, and water and sanitation. Currently they have a staff of about 40 people, most of which are what they call surveillance officers who spend most of their time in the field working with local volunteers. Other staff are administrative staff, program managers and researchers.

Their work on malaria prevention focuses both on research on malaria transmission, community surveillance and indoor residual spraying (IRS). To help them in this work they have, together with external developers, created a software system, called mSpray, in which they use geographic information system tools (GIS) such as QGIS together with Google Earth to enumerate houses and structures in satellite images, and mobile-cellular data enabled Android tablets with data collection tools (Open Data Kit) to collect data from spray teams. This enables real-time collection of data and the spray teams use the map and GPS functionality of the tablets to see exactly where they are and where they need to go next. In addition to this, a number of CHWs has been trained to test for and treat malaria, and use feature phones to submit reports on malaria cases to DHIS2, enabling early detection of malaria outbreaks.

Working together with the Zambian Ministry of Local Government and Housing and UNICEF Zambia, Akros' water and sanitation programme (WASH) aims to eliminate food and water contamination by working with the local communities to educate them on sanitation and hygiene and encourage them to build proper latrines and hand washing stations. Local volunteers, which Akros calls Community Champions (CCs), are their feet on the ground and use feature phones to submit monthly reports on water and sanitation status in the villages and districts. The surveillance officers conduct regular visits to each district to assess the work being done by the CCs and the district officials and assist them with problems they might encounter. Due to the focus on working together with the local communities, they have dubbed this project Community Led Total Sanitation (CLTS).

Several Akros staff members have gained considerable experience working with DHIS2 in connection with their own programmes, and they have recently started offering DHIS2 training courses for governments and organisations and have seen a high demand for these trainings.

7.3 ZCAHRD

The Zambia Center for Applied Health Research and Development (ZCAHRD) is an NGO affiliated with the Center for Global Health & Development at Boston University (BU) who does research on critical issues of public health and serves as a partner to the Government of Zambia in health service delivery, formalized through a Memorandum of Understanding (MOU) with the Ministry of Health (MOH). They also work closely with the Ministry of Community Development, Mother and Child Health (MCDMCH), the ministry responsible for all primary health care services as well as maternal and child health in Zambia. ZCAHRD is led by a BU faculty member native to Zambia and has 35 permanent employees based in Lusaka and Choma, and also employ a number of temporary employees and contractors such as translators and data collectors. One of their main goals is to generate evidence that can support health policy decisions that will improve the health and wellbeing of the general population in Zambia.

They have completed and active programs in a number of areas, such as monitoring and evaluation of public health programs; research on malaria treatment and prevention; maternal, newborn and child health programs as well as many other.

Chapter 8

Method

This chapter will cover the methodologies I based my research approach on, namely Action Design Research and the interpretive case study. I will explain how and why the specific case was chosen, before giving a summary of the data collection methods employed and finish of with a discussion and review of the

8.1 Research Method

The research methodology this thesis is based on can perhaps best be described as a case study with elements of participatory action research and is based on qualitative, interpretive research.

8.1.1 Action Design Research

Action Design Research is "a research method for generating prescriptive design knowledge through building and evaluating ensemble IT artifacts in an organizational setting" and builds upon the Action Research approach.(Sein et al., 2011) Action Research can be viewed as a cyclical process with five phases: diagnosing, action planning, action taking, evaluating and specifying learning, where all five phases are considered necessary for the definition of action research, but the number of phases actually carried out in collaboration between researcher and client organisation may differ(Susman and Evered, 1978). The original intention was that I would focus on Akros as a case study, and try to identify areas in which they could improve and make suggestions for improvement and analyse the results of any changes made to their work process. But then the opportunity for contributing to the iCCM study came along and the way I was thrown into it made it difficult to plan any proper research approach. I therefore cannot say that the ADR method was followed as well as it should, and my work took more on the form of a pure participatory case study where I participated in the work on equal foot with those I worked together with.

8.2 Selecting a case

After decicding that the chosen overarching topic for my thesis would be ”Health information systems in developing countries”, it took quite some time before a specific case could be chosen as a subject for study. I would work on this thesis in cooperation with the HISP group at UiO, and early on there were some indications that HISP UiO would be awarded a contract with UNICEF to work on a rather large development project in Zambia, and it was decided that if their application went through, that I could find some role in the introductionary phases of that project. The process dragged on however, and everything was still in limbo as spring came around and the summer approached, so since things started looking a bit bleak for the UNICEF project we decided in late April that I would go to Zambia in the end of May to attend the DHIS2 Academy that would be held in Livingstone in early June. But even at that point, late April, it was not fully decided specifically what my case would be, but there was a possibility that I would be able to join up with Akros, a Lusaka-based NGO that HISP UiO was interested in building a relationship with because of their novel uses for DHIS2. Thus, the decision to choose Zambia as a location for field studies was twofold. Firstly, the DHIS2 Academy would present a great opportunity to gain insight into how DHIS2 is used by governments and NGOs in South-East Afria, as well as giving me an introduction to the software itself which I was not previously familiar with. Secondly, the possibility of joining Akros and study how they The actual agreement with Akros was not made until during the Academy in Livingstone, where it was agreed that I would join Akros at their office in Lusaka and work with them for the rest of my stay. The intention was not only for this to be a case study where HISP and myself would benefit from studying Akros and their use of DHIS2, but that they would themselves potentially benefit directly from it by having me help them out in their work as much as I could which they looked forward to since they currently did not have any staff with background in IT.

I joined up with Akros a few days after the DHIS2 Academy ended, joining them first on a District Level DHIS2 training in Mazabuka district before heading to Lusaka. After spending a few days at their office, I was told that they were considering taking on a consulting job for another NGO, the Zambian Center for Applied Health Research and Development (ZCAHRD). ZCAHRD was in the process of conducting a study on an mHealth intervention in Integrated Community Case Management (iCCM) and had decided to use DHIS2 as the software system to base this intervention on. They had been in touch with HISP UiO on this matter, but since no one at ZCAHRD had any previous experience with the software they needed the help from a third party in order to configure and implement the system. HISP UiO had therefore contacted Akros, who they knew had solid experience with DHIS2 from their own projects, and asked them if they were interested in taking on this job for ZCAHRD. Akros is a small NGO with little manpower to spare, and they initially decided that they most likely could not take the job because they did not have the resources to do it, but since I would now be spending a couple of months working with them, they asked me if it was possible for me to join them on this project and have that as my subject of study. In that case I would mostly have to drop my original plan on doing a case study on Akros the organisation, and instead do a case study on the implementation of DHIS2 for use in iCCM mostly based on already decided on requirements and specifications. What made Akros interesting to begin with was their work on water and sanitation, their mSpray malaria prevention and elimination programme, and their intent to use DHIS2 in the education sector. The malaria programme was in a period of low activity, due to it being winter in Zambia, and Akros would not do any field work themselves before late August/early September and funding was not yet in place for the education project so no work would be done on that just yet. This left me with the CLTS project, where they had no new districts or chiefdoms coming on board so they were just doing supervision visits of the kind I had already briefly attended. All this, together with the fact that HISP UiO was very interested in the iCCM study and delighted at the fact that I would have the opportunity to work on it, influenced my decision to drop my original case and assist Akros so that they could take on this project.

Now, much of this background information I have just presented the readers with, is just to communicate the fact that the circumstances I found myself in throughout my work on this thesis did not always give me the best opportunities to plan beforehand how I should go about my research. I knew I was going to Zambia, I knew how long I was going to stay there, but it was not until *after* arriving there that I knew more specifically what I would study and who I would be working with. And then there came a switch of focus from studying Akros as a whole to focusing on the particular instance of working with them on the iCCM project, so decisions was sometimes made without being given too much time to think through. While this leaves some weaknesses in my overall approach, such as a lack of field trips and not completing the cycle that is so central to ADR, sometimes when the opportunity presents itself you just have to grab it and then make the best of it as you can.

8.3 Data Sources

During the literature study, I employed searches in library databases to find relevant literature using search phrases and terms I deemed appropriate. When articles with useful information was found, I would look through the bibliographies to find the articles that the more interesting citations were taken from, and to see if there were any promising titles of papers and articles that could be worth checking out. Published articles and papers were the main source of information, but for numbers on things like mobile subscriptions and information on Zambia I turned to reports from well renowned organisations such as GSMA Intelligence, WHO and UNICEF. The chapter on HISP and DHIS2 is mostly based on information I have gathered while attending meetings with HISP UiO, conversations with members of the core HISP team and the DHIS2 Academy in Livingstone. Two prominent members of HISP UiO have published a book about the program, which was my main source for the HISP history section. While there is always the risk of bias when the main source is the subject in question, I have judged it to be an acceptable approach since it is not HISP that is the focus of my study and as such does not influence my results.

During my stay in Zambia, the DHIS2 Academy was a main source for information about the HISP organisation and also gave me a good introduction to DHIS2. I participated in the Academy as a regular participant, working together with representatives from various countries and organisations on group assignments and workshops as well as conducting some informal interviews. Being new to DHIS2, and not being familiar with the terminology it uses or its advantages and disadvantages, getting into any technical discussions with other participants was difficult, so I mostly concentrated on getting a broader overview of who used it and for what as well as getting an idea of how the different nodes in the HISP network worked together.

Before the Academy started, I spent a day in the field with a representative from Akros and some members from HISP UiO, where we got an introduction to their Community Led Total Sanitation project (CLTS). This took place in the Kazangula district west of Livingstone, and included meetings with some district officials as well as a few of the volunteer Community Champions (CCs). A change in circumstances, which will be discussed later, led to this being the only "real" field trip during my stay, if we define a field trip as getting down to the ground and meeting with the individuals doing the actual work in the communities. It was a very interesting and worthwhile experience, even though it turned out not to contribute anything of value as such to my continued study.

Later, I would spend a few days participating in a District Level DHIS2 training that Akros held for district officials in Mazabuka district, a few hours south of Lusaka. This was quite a contrast from the Academy in Livingstone, with only six participants, and had more of a classroom-feeling with a great deal of interaction between the Akros trainers and the course participants.

During the rest of my stay, my main data sources were as follows:

Field notes, which I would take during every meeting and during work sessions if something stuck out as being out of the ordinary. They are a collection of my thoughts, observations and noteworthy comments picked up during my entire stay.

Meetings and conversations I participated in meetings between Akros staff before they decided to accept work on the iCCM project, as well as the initial meetings between Akros and ZCAHRD when the details were worked out. I was more or less treated as one of their own employees, and learned much about Akros and their work from just working and socializing in their offices

Observations Working closely with Akros staff for about two months gave me lots of opportunities for observation, but as I was perhaps a bit too focused on working on the DHIS2 implementation I rarely took the time to step back and just observe.

Electronic communication With a distributed team, we often had to resort to electronic communication with team members in the form of email conversations and Skype calls. Communication with the developer in Vietnam and the hosting company in the US was done mostly by email, but we did have a couple of Skype calls with the developer. Any email sent to the developer or hosting company would be CC'd to the other Akros team members so that everyone was kept in the loop on the current status. One member of the ZCAHRD staff created a shared folder for all of us on Google Drive so that it was possible to cooperate on the writing of training materials and so called surveillance protocols.

I myself did not participate actively in that part of the work, but I had full access to the shared folder and could see what was being worked on by whom.

Other sources include various documents related to the iCCM study provided to me by ZCAHRD and Akros.

Everyone I worked with was made fully aware that I was there as a student working on my master thesis and the work done would be the focus of my research.

8.4 Analysis

As stated, the Action Design Research approach places emphasis on the cycle. Unfortunately, I was not able to complete even one iteration of this cycle due to the nature of the case that was ultimately selected. The literature study I conducted before traveling to Zambia was not focused on this specific case, but on the broader topic of ICT4D since I had yet to choose a case at the time. Information about iCCM, the requirements and intended purpose of the software system was already compiled and decided on by ZCAHRD. The software artifact that was produced, namely the DHIS2 installation, was put to use too late for me to get any data from it. I had hoped that I would get about six to eight weeks worth of reports from the CHWs which I could then potentially use to form some vague, inital impressions of the immediate effect of the intervention. However, it turned out that the project got delayed somewhat and left me with only two weeks worth of data, which I had no time to look at properly. This study deals therefore entirely with qualitative data.

However, in addition to the software artifact that is the iCCM DHIS2 instance, some of the training materials I produced (see appendix) are being used ”far and wide” as I was told in my last conversation with my contact in Akros, and if that is the case I consider it reasonable to assume that they have been considered useful for training purposes.

There was also the creation of a step-by-step tutorial for installing and running DHIS2 on a Virtualbox virtual machine running Ubuntu, and also the beginnings of a tool to easily launch and manage seperate DHIS2 instances on that virtual machine for use during trainings so that Akros could conduct trainings without needing to relay on internet access. This work was started while we waited for confirmation that the iCCM project was good to go, so it got abandoned halfway through and has not been properly evaluated. Even so, it has to my knowledge had the chance to prove its usefulness on at least two occasions: once during an Akros-led DHIS2 training in Botswana where the online server went down and they decided to test the local server while waiting for it to come back online, and once during a training in Nigeria. Akros has purchased a few portable computers (Intel NUCs) with the intention of using them as local servers during trainings, so it may seem that my efforts to show them how relatively easy it is to set up a machine, virtual or otherwise, running Ubuntu and installing DHIS2 has encouraged them to try new things. Depending on the time and effort they put into this, it might lead them to become comfortable with doing more server administrative work and thereby lessen their dependence on their hosting company and consultants when problems occur.

When doing case studies there is always the possibility of the collected data being interpreted wrongly, or that prejudices of the researcher influences the conclusions drawn. As far as I can tell, I have treated the subject in an objective manner and spent much time discussing the work being done both with ZCAHRD and Akros staff that I worked together with, as well as the student interns that worked at Akros while I was there. These discussions helped me get a different view on things and a varied set of opinions.

Chapter 9

Case

This chapter will cover the specific case that I worked on, which was to configure and deploy a DHIS2 installation to be used in an iCCM mHealth intervention iCCM study ZCAHRD was conducting in cooperation with UNICEF and the Zambian Ministry of Health. I will first give an introduction to iCCM and the rationale behind the study, before discussing the requirements for the iCCM DHIS2 instance, the work I did on this project together with Akros and ZCAHRD staff.

9.1 iCCM

In Zambia and other DCs, there is often limited access to health care in rural areas and treatment is often complicated by shortages of the required medicines and understaffed health centers. Many countries have been experimenting with community-based programs to address this access gap. A common approach is to give volunteers from the local communities training in basic disease diagnosis and treatment and have these community health workers (CHWs) be the first point of contact for health care in rural communities.

Three quarters of deaths among children under age five years is due to a handful of causes: pneumonia, diarrhoea, malaria and newborn conditions. The correct diagnosis and treatment of these three diseases is therefore essential for reducing childhood mortality. There is a significant overlap in the clinical symptoms of these diseases, and an integrated strategy for providing care and treatment is seen as highly desirable. This has led to increased adoption of integrated community care management (iCCM) where CHW's are trained specifically to diagnose and treat these diseases in an effort to extend the reach of public health services and providing timely and effective treatment. When visiting a sick child, the CHW screens the child for the three diseases and administers the proper treatment. They will also advise the mothers on proper home-care and provide them with neccessary medicine, oral rehydration salts (ORS) and zinc. CHWs are supplied with a small stock of neccessary medicine and supplies; artemisinin-based combination therapy (ACT), oral antibiotics, oral rehydration salts, zinc and rapid diagnostic tests (RDTs). Antimalarial drugs are expensive, and incorrect use can cause the current drugs to become less effective so it is a WHO guideline that all suspected malaria cases be confirmed with RDT before beginning treatment.

There are several benefits to iCCM, including giving children early access to treatment, reduced use of antimalarial drugs because of the RDTs, reduced workload at the health centers and probably a decrease in mortality amongst children under age five, although that is still uncertain. Earlier studies have also shown an increase in the proportion of mothers seeking care from CHWs and a decrease in use of health facilities and traditional healers.

A previous study in Zambia showed that with training and supervision, CHWs can correctly classify and treat pneumonia and malaria but a study from Uganda and one from Kenya showed some problems with evaluation of pneumonia and placed emphasis on the need for continuing supervision, training and quality measurement of CHWs performance.

While iCCM is considered to be a valuable strategy for extending the reach of public health services, it still faces a number of challenges, including efficient supply chain management, appropriate supervision and effective scaling strategies. A previous study in Zambia showed that with training and supervision, CHWs can correctly classify and treat pneumonia and malaria but a study from Uganda and one from Kenya showed some problems with evaluation of pneumonia and placed emphasis on the need for continuing supervision, training and quality measurement of CHW's performance. Wazny and colleagues have identified 20 priority research questions for iCCM (Wazny et al., 2014), and Boston University is in the process of conducting a study, acting through ZCAHRD, focusing on three of the research questions: (1) evaluation of strategies to improve supervision and quality of care using mHealth technology, (2) evaluation of strategies for, and costs of, supervising the CHW and (3) evaluation of strategies to improve integration of iCCM logistics (diagnostics and drug supply) to the central procurement and supply system at the community level.

9.1.1 Study Design

The study is designed as a cluster randomized controlled trial, where the cluster is defined as a health center/health post, including all the CHWs providing iCCM in its catchment area and all children under age five with malaria, pneumonia or diarrhoea that have been treated by a CHW. Half the clusters will follow the current standard MCDMCH/MoH iCCM practices, the other half will have enhanced inventory management and supervision using mHealth interventions. They are designated as control and intervention arms, respectively.

The study is to take place in Chipata and Chadiza district in the Eastern Province. From the two districts, a total of 40 clinics and health posts were selected at random, 30 in Chadiza and 10 in Chipata, and half of the selected clinics in each district was randomly assigned to the control arm and the other half to the intervention arm. For each clinic, one CHW and one member of the clinic staff was selected for participation in the study. Necessary training for ZCAHRD, District and clinic staff and participating CHWs was originally scheduled for late September/early October and data collection would start as soon as the trainings were finished. Unfortunately, the trainings were delayed for some weeks and did not get underway before the start of November. More details on this will be provided in the next chapter. The study is scheduled to last until June 2016, and based on the results a decision will be made to scale the system nation-wide, something for which funds have already been secured.

9.1.2 Disease Reporting and Inventory Management

The intervention will focus on two important aspects of iCCM: commodity management and disease reporting. CHWs keep small stocks of medicine and equipment needed for diagnosing and treating the three diseases, and the current system have them keeping track of commodity levels by filling in a paper based form once a month and submitting it to the clinic. They usually get resupplied once a month, but can request an extra restock if they have run out, or is close to running out, of an item long before the next restock is to take place. Restocking is done at the health posts and clinics, and the CHWs have to travel to their respective clinic to collect the supplies.

Once a month, CHWs are supposed to submit reports on disease statistics once a month. Currently they have a ledger which they fill in whenever they treat or see a patient, and at the end of each month they summarise that month in a section at the bottom of the form. The data collected consists of the total number of cases for each disease, how many of those cases were children under the age of five, how many of those were male, how many female, how many children were referred to clinics and how many children died. Any patients the CHW judges too ill to be successfully treated by the CHW, should be referred to the nearest clinic to receive treatment there.

When discussing these two separate parts, I will use the term Request and Requisition (R&R) for the commodity management, and disease management for disease statistics and referrals as these were the terms we used when I worked with Akros and ZCAHRD. For this study, the intention is to keep using the paper based forms, but have the CHWs submit the data via mobile phones on a weekly basis. Dedicated staff on all levels in the hierarchy, from clinic staff through district up to ZCAHRD and MoH will then be able to look at data for both commodity usage, and disease cases in near real time compared to the current situation where only a very limited group of people get to see the numbers. For the R&R part, the hope is that this higher frequency of reporting, combined with the information being available to multiple levels in the supply chain hierarchy, will contribute to improved supply chain management and reduce stock-outs. When it comes to disease management, the higher reporting frequency will allow for more accurate statistics on the different diseases and perhaps also enable one to uncover hidden patterns in how and when diseases spread. It is, for instance, well known that the prevalence of malaria varies with the seasons. In addition to the reports being submitted, a mobile application will allow the CHWs to send advance notice to the clinics when referring a patient. The way it works now, is that a patient just shows up at a clinic with a referral slip from the CHW, but the clinic might be totally unprepared for e.g. a patient with a severe case of malaria and the clinic staff may have to suddenly re-priortise their work. The hope is that when the clinic gets notified instantly when a CHW refers a patient to them, they have enough time to do the necessary preparations, since the referral is supposed to include a preliminary diagnosis made by the CHW so the clinic knows what they will be dealing with.

9.1.3 Requirements

There were a number of requirements that a DHIS2 implementation had to satisfy for it to be suitable for the iCCM intervention. Most of these were already in place in the default DHIS2 that is distributed by HISP, but some of the requirements needed some extra development. The complete requirements document is included as figure 11.1 in the Appendix.

9.2 DHIS2 configuration

DHIS2 was chosen as the as the software system to base this intervention on, and most of the required functionality was already supported by DHIS2, but there needed to be done some modifications to both the mobile applications and on the server side for all requirements to be met. Akros had been in touch with HISP Oslo and through them enlisted one of the HISP Mobile developers, located in Vietnam, to help with developing the extra functionality. Also brought in to help was an external consultant, located in Sweden, whom Akros have used previously in their own projects and who has extensive experience with DHIS2 server configuration and the development of HTML-reports. The DHIS2 instance itself was hosted by an American company who specialises in DHIS2 hosting and Akros has had good experiences with them, using their services for all their own DHIS2 instances. All of this means that while Akros was the implementing partner, they outsourced critical parts of the work to form a highly distributed team working in three different time zones. As we shall discuss later, this arrangement brought with it some problems and delays.

Configuring DHIS2 to be able to support the iCCM intervention required not only creating converting the paper-based forms to DHIS2 data sets made available to CHW's via the mobile applications, but also creating the necessary user roles and organisation units in the organisation hierarchy. Akros already had access to a DHIS2 instance containing the official complete organisation hierarchy for the Zambian health care system, so it was just a matter of getting permission from the MoH to export it and then import it into the iCCM instance. In the existing hierarchy the health centers/health posts are listed as being the lowest level, so in the iCCM instance we needed to create an extra level to accomodate the CHW's. However, CHW's are individual persons and an organisation unit should not represent a person but an organisational entity which individual users of the system are then assigned to We decided to call the new organisation unit for Community Health Access Point (CHAP) which would then represent more or less the area in which a CHW is located. If a CHW quits or decides to move, and a new CHW is brought in to cover the same area there is no need to make changes to the organisation units since the new CHW is then just assigned to the CHAP in that specific area. This solution is however just a quick fix to something neither ZCAHRD nor Akros had thought about before we encountered the problem of representing CHWs in DHIS2, and it was deemed adequate by all parties for the sake of this study, but it was agreed that this needs to be further examined if the program is to be scaled nation-wide.

Different users of the system need different levels of access, so there needed to be different user roles for every level in the hierarchy. When creating user roles in DHIS2, they need to be given a set of authorities in the system which define which data and functionality in the system users assigned a specific role has access to. We found there was a severe lack of proper documentation for the different authorities, of which there is quite a selection, so instead of trying to figure out what each of them granted access to we decided to again just copy the user roles from an existing DHIS2 instance since creating them from scratch was not considered necessary and would have been mostly a waste of time for the purpose of this study.

Creating data sets from the paper-based forms was mostly straight forward, but the size of these forms caused some concern since Akros has experience from their own programmes that large forms leads to a drop in reporting rates. More specifically, I was told that they had seen reporting rates drop once forms became longer than 25 data entry fields. The iCCM disease reporting form has a total of 48 fields which are to be filled by the CHW every week, which can therefore easily be seen as way too many. This problem was explained to ZCAHRD, and we discussed the possibility for reducing the number of fields in the form in order to try and avoid incomplete reporting rates but that turned out to be difficult. According to ZCAHRD every field was more or less equal in their importance and relevance for the study so there were none that could be removed from the form. The form was designed as a matrix, with seven columns and eight rows and, it was easy to see that one of the columns could be replaced by an automatically calculated indicator, since it was just the sum of two others. This enabled us to reduce the data entry field count by seven. By removing the fields dedicated to other diseases, that are not part of iCCM and replacing them with fields for newborn care management, got rid of yet another three fields, brining the total down to 38. Please see figure 11.4 for the paper-based form and figure 11.5 for the one implemented in DHIS2. Both figures can be found in the Appendix. The same applied to the R & R form, which had even more fields on the paper form, but allowed for more of them to be replaced by calculated indicators. See figure 11.2 for the original form and figure 11.3 for the DHIS2 version.

When working on creating the data sets in DHIS2, I accidentally discovered that the forms could be divided into sections which allowed us to group related fields together. Akros was unaware of this possibility so they had so far been designing the forms as just one long list of data fields to be filled. This can be challenging to fill out correctly when doing it on paper, and would be even more challenging to do on a small phone screen where you cannot see the entire form at once. By dividing the forms into sections, where each section corresponded to a row on the paper based form, it will hopefully make it easier, and less frustrating, for the CHWs to fill the form on their phones and also reduce the probability for making mistakes since it will be easier to check the numbers entered on the phone against the ones on the paper. Having the form divided into sections increases the navigational complexity on the phone however, but our guess is that the advantages outweigh the disadvantages.

The Patient Referral Tracker is completely new to iCCM and has no analog counterpart so it had to be designed and created from scratch. As already mentioned, the basic idea behind the tracker is to enable clinics to be notified when CHWs refer patients to them, and it will also allow clinic staff to give valuable feedback to the CHW, e.g. if the patient was diagnosed correctly, if the treatment given was wrong etc. In DHIS2, the tracker module lets you define so called *"tracked entities"*, which can be any *"thing"* one would want to track over time. For example, Akros uses the tracker to track things like water points (wells, bore holes etc), but in the case of the referral tracker we would like to track patients, so we defined the tracked entity to be of type Person. A tracked entity has a certain set of attributes which do not change over time, e.g. name, date of birth, and gender. When starting to track an entity, one first enrolls it in a tracker program, and then track it through two or more program *stages*, with each stage taking place at a different point in time. The stages can be different from one another, or it can be a single stage repeated at regular intervals. For the referral tracker, we wanted two seperate stages which the patient would go through: the referral stage, where the patient would be registered in the tracker program when the CHW refers the patient to a clinic, and a discharge stage, which the clinic staff would complete when a tracked patient is discharged from the clinic. When explained how the DHIS2 tracker functionality works, and that the program can be expanded at a later time to include additional stages, e.g. referring a patient from a clinic to a hospital, ZCAHRD expressed interest for exploring the possibilities this provides.

ZCAHRD also wanted the possibility for clinic staff to submit monthly reports on any problems faced during the mentoring of CHW's. Since one of the goals of this study is to improve the training and mentoring of both CHW's and clinic staff, this is an important tool to collect valuable data which can be used to identify which areas the CHW's struggle with the most. This specific requirement had managed to avoid attention from both Akros and ZCAHRD and it was not until very near the end of my stay that it resurfaced and received any attention. When conducting mentorship sessions, clinic staff have a checklist they follow and use to note down what the CHW's are doing correctly or incorrectly, consisting of checkboxes and fields for remarks and comments. The paper based checklist contains a lot more information than what is realistic to expect them to enter on a mobile phone, and ZCAHRD was mostly interested in basic aggregate data so it was decided that the monthly mentorship report would just consist of selecting the top-three issues encountered during that months mentoring, from drop-down lists, a text field belonging to each issue to allow for comments, and a fourth text field that could list any additional issues not covered in the pre-defined list.

There was also the requirement that DHIS2 send out automated SMS to clinics when CHWs registered a new patient in the patient tracker, as well as some SMS reports and reminder messages. All work on these features were done after I left Zambia, so I had no involvement in them, but I have been told that they were all completed and fully functional by the time the trainings started.

9.2.1 Training materials

As part of the agreement with ZCAHRD, Akros was supposed to train key ZCAHRD staff members in the use and configuration of DHIS2. Since we would be working closely together with the iCCM program manager, the intention was to train him in DHIS2 usage while we worked on configuring it; showing him what we did and how to do it. Later on, after the implementation stage was finished, Akros would hold a training session for a small group of ZCAHRD staff.

After about a week of the Akros HISM and me working on this, Akros got some new staff members in the form of four Global Health Corps Fellows (GHC Fellows), and two of them were to join us on working on the iCCM project. This meant that the workload would be divided on more people, but also an increase in work since they would have to be trained in using DHIS2, having no prior experience with it. Not only that, but ZCAHRD got their own GHC Fellows, one of which were to work on the iCCM project. So now our numbers doubled, from three to six people, and instead of teaching one person how to use and configure DHIS2, there were now four people without any experience with the system that needed to be taught how to use it at the same time as we worked on configuring it. When starting working on this project, I had no prior experience with DHIS2 except from what I picked up at the Academy, but I was soon tasked with being responsible for training the rest of the team while at the same time teaching myself. To help with the training, I created some figures and diagrams, which are available in the appendix, showing how the different elements that make up a data set or tracker program belong together and depend on each other. This helped both for reinforcing my own understanding of how the system works, as well as making it easier to communicate across language barriers. We were a mix of Americans, Norwegians and Zambians, so even though we all spoke English, we differed in our level of command of the language.

Some examples of the training materials I created can be seen in the Appendix.

Surveillance and Supervision Protocols

Something that was not part of the requirements, but which Akros considered necessary, is what Akros calls Surveillance and Supervision Protocols. This is something they have developed themselves and chosen to use in all of their own projects both to make trainings easier and to give their field workers a document to consult if needed. Basically it is a set of instructions covering everything that needs to be done during surveillance and supervision visits. The protocols should be detailed enough so that ”everyone who can read, should be able to perform the work.” A set of protocols were developed for the iCCM study, which were largely based on the existing protocols that Akros uses, but I did not take any direct part in working on them.

Akros is of the opinion that these protocols are of great help, since they provide a complete checklist for all the work that should be carried out during supervision and surveillance visits. However, they have done no proper assessment of their effectiveness and when I asked them about it, they confessed to having some concerns about the protocols leading to rote learning of the work to be performed instead of letting the workers learn to understand how and why to do it. If you always perform your work by just mechanically following steps lined out in a checklist, then you never gain the understanding needed to deal with anything that falls outside of that list. Akros expressed interest in having the use of such protocols properly studied to decide whether they provide the benefits they assume they do, but they do not have the time or resources to do it themselves.

Part III

Discussion and Conclusion

Chapter 10

Discussion

This chapter will discuss some of my experiences from working with Akros and ZCAHRD and explore some of the problems that were encountered, both of a technical kind and of the organisational kind and end with some recommendations I would make to Akros should they ever find themselves with a similar case.

10.1 Planning and Development

The software development methodologies I discussed in a previous chapter, the Waterfall and Spiral model and Agile development can also be applied to a project such as this. We could, and in retrospect probably should, have spent some time analysing the task ahead of us and choosing a suitable approach, but we did not and we were worse off for it, in my opinion.

The project as a whole suffered somewhat from a lack of proper planning and not having a proper roadmap to schedule the work to be done. Akros had provided ZCAHRD with a Statement of Work narrative (SOW) and a list of deliverables with expected delivery dates, but this was only loosely followed. Several times we had agreed upon an agenda for a meeting with ZCAHRD, and then that agenda would be disregarded and we would spend hours doing something completely different. A specific instance was when the ZCAHRD representative had some questions for us after attempting, and failing, to create a tracker program. Instead of working together on the supervision and surveillance protocols, which was the original purpose of the meeting, we instead spent several hours discussing how one could go about creating a tracker program for a fictional childhood vaccination program and then walking the ZCAHRD representative though each step in implementing it in DHIS2. While this diversion was valuable in its own way, since we were supposed to train ZCAHRD staff in how to use DHIS2 after all and it gave the GHC fellows their first introduction to the DHIS2 tracker functionality, it still disrupted our plans for the day. The Akros HMIS advisor was clearly annoyed about this since it delayed our work, but he accepted that this was not always that easy to avoid, saying that ”We need to keep him [ZCAHRD rep] happy, otherwise this is not going to work. You don’t start working together and then become friends, you need to become friends first and then you can do work together."

The question of how to conduct the training of CHW's was not discussed during any of the initial meetings that I attended, and it became apparent in a Skype call we had quite late in the process that Akros had assumed that ZCAHRD would let Akros design the training format, seeing as they have considerable experience in the area, running approximately 30 trainings per year for their CC's. Since ZCAHRD had scheduled two weeks for training of CHW's, Akros had planned on doing a centralised training for all the CHWs, the same way they do it when training their own CC's. They tend to gather all the CC's together on a per district basis in order to train them all at once over the course of several days. In their experience this is a good way of doing it since the CC's need time to become familiar and confident in the use of the mobile applications and that one or two days is too little time to cover all the topics. But ZCAHRD protested and said that this way of doing it was something that was simply not in their budget. It was not just the cost, but also the logistics behind providing transport, food and accommodation for approximately 60 CHW's would prove too challenging on such a relatively short notice. They would much prefer to send ZCAHRD staff from district to district in order to train the users, thereby keeping the cost down by only having to cover expenses for two or three staff members. Akros argued against this approach, citing their own experiences and pointing out that since the success of the intervention is dependant on the CHWs being able to use the apps then "... if we can't get the training right, all work until now will be useless." ZCAHRD had previously told us that funding for scaling this nation wide was already in place if it showed positive results, and that one of the main issues would be "... to show that reporting from CHW's will actually happen." But without proper training, the argument went, neither the CHW's themselves or the clinic staff would be able to do much troubleshooting if problems occur and thereby possibly reduce reporting rates. In the end, the one-on-one trainings were chosen because there simply were not enough funds for anything else.

An Inexperienced and Distributed Team

While the work was originally scheduled to commence in early July, some disagreement about payment needed to be worked out before we could start and led to a delay of about two weeks. When Akros decided to take on the job, after I agreed to work with them on it, we knew that the Akros HMIS adviser whom I would be working together with had to leave for the US in the end of July and would be away from Zambia for about a month. Even so, we estimated we would have roughly four weeks to work on it before he left, which was assumed to be more than enough time to finish the configuration of DHIS2 and leave mostly things like testing the system and training ZCAHRD staff to those of us staying in Lusaka. The delay caused by the payment issue meant that his time available to work on the project from Zambia was basically cut in half, down to two weeks. When he left, it meant that the two GHC fellows and myself were more or less in charge, with none of us having any prior experience from similar kind of work to draw upon.

This put us in a somewhat difficult situation. I was there as a representative from NTNU and UiO and had nothing but an informal agreement with Akros, so I was in no position to make any decisions on their behalf, and the GHC fellows were looking to me as the more experienced person on the team even if that extra experience only amounted to two extra weeks working on this. So when the issue of training format cropped up, none of us could realistically argue against, or agree with the ideas ZCAHRD had for the training format. We all knew that Akros' plan was to get all of the CHWs together and do a week-long centralized training, but having no experience from those kinds of training we could not make any authoritative arguments when ZCAHRD presented their plan to travel around to each district and do one-on-one trainings.

Mobile app issues

The lack of proper planning became evident also in the constant iterations of the mobile apps. During our initial Skype call with our developer, we discussed the need for some modifications that needed to be made to the J2ME apps. Most required functionality was already supplied by the apps, but on the list of DHIS2 requirements for the iCCM study, there were some requests for added functionality: By default, the aggregate capture app displays a list of days to choose from when a user is to submit a report, allowing the user to choose any date and then DHIS2 will make sure that the submitted data is added to the correct reporting period. This was requested to be changed from listing every day to listing week numbers, since the reports were to be submitted weekly. There were some protests from the developer about this since ”... week numbers might not be known to everyone...” and it would require some more than minor changes to the DHIS2 back-end which does not operate with week numbers. Instead, he proposed to use date ranges, e.g. 05.09-11.09, so that users need not keep track of week numbers and less changes were needed to the back-end. It seemed to me that the largest change that needed to be done to DHIS2 and the mobile apps to fulfill the requirements, was the requirement that DHIS2 should automatically send an SMS notification to a clinic when a patient had been referred, but this was a requirement addressed only very near the end of the project and seemed to have been forgotten by everyone involved until that time.

It turned out that there were all sorts of small issues with the apps that, at least to me, was quite unexpected since they were not mentioned in our initial conversation with the developer. For instance, when using the tracker app not all data fields would we displayed in the application and even though fields marked as ”mandatory” were not filled in, the app was still able to submit a registration and it would show up in the DHIS2 database. In both tracker and aggregate app, data fields configured as drop down lists did not display as such and let users enter arbitrary data, which is illustrated in figure 10.1. The aggregate app would also fail in updating previously submitted reports and the data submitted would not show up when accessing DHIS2 via the browser. We informed our developer of these issues as soon as we discovered them, but because we did not have a proper plan for testing the apps we did not report all of the issues all at once so it took several iterations before all the kinks were ironed out. Had we taken a risk-based approach, and spent some time in the beginning to identify and evaluate the risks that could affect the project we would probably have identified the mobile apps as a high risk element. After all, the entire study

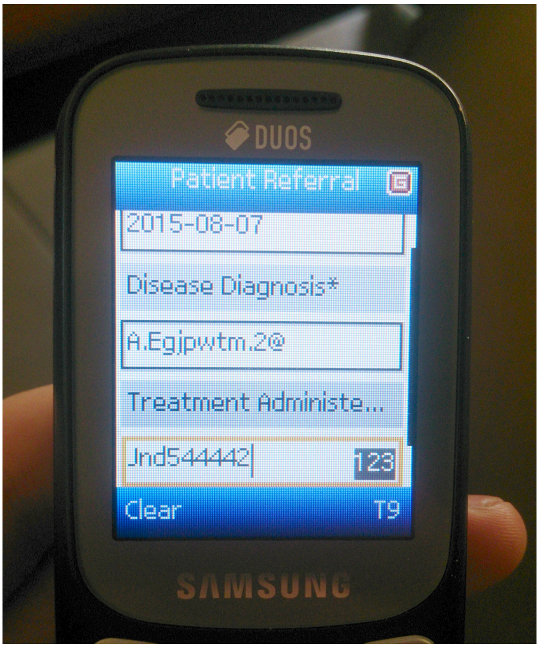


Figure 10.1: Fields erroneously allows entry of arbitrary data

relies on the CHWs being able to report data via their mobile phones and any delays with the apps would cause a delay to the entire project.

The issues with the mobile apps might have been resolved more quickly if we had had an on-site developer which we could communicate with directly instead of having the long delays associated with being in different time zones. The same was true with having the DHIS2 instance hosted in the United States. Time zone differences would often led to long delays when an issue cropped up that needed to be solved or looked into by a different party than the one who discovered it. For instance, when we needed to change the server configuration so that it defaulted to HTTP instead of HTTPS, we needed to communicate this request to the hosting company which was eight hours behind. We received their reply the day after we sent the request, where they pointed out that HTTPS would mean insecure data transport and asked us if we really wanted to do it. Our answer to that, saying that yes we really were sure, did not reach them until the end of our working day. This caused it to take almost two days from we discovered the problem to having it resolved, instead of having it solved the same day as would probably have been the case had the server been hosted in the same time zone. Delays like this regularly occurred when we received a patched DHIS2 file from our developer that needed to be deployed on the server, usually we could not test the new version until the day after we had requested it to be deployed.

In short, the problems with the mobile apps were to some degree the result of poor planning and lack of time. Akros had previous experience only with the aggregate reporting app and had never used the tracker app and was unaware of the amount of work required to make it work for this project. Had we taken some time in the beginning of the project to test the apps and figure out exactly what functionality was missing or not working as wanted, we could have specified more clearly to the developer what he needed to do which would most likely have led to him giving us a more correct estimate of the time needed to make the necessary changes.

When reviewing the way we worked, how we started out with a broad overview of what was needed but did not have all the details worked out about the different aspects of the system, leading us to an iterative development of both the DHIS2 instance and the two mobile apps, it can perhaps be described as hybrid approach to system implementation, with elements from both Agile methods and the Waterfall model. But truth be told, we did not try and follow any established methodology or best practices and generally just agreed from day to day what we specific part we should work on. Considering the detailed specifications ZCAHRD had worked out for the intervention, and the fact that we had a deadline not too far away and knowing that slow internet speeds meant that creating all the required data elements and organisation units and so on in DHIS2 would take quite some time so it would be best to get it right the first time, I would say that a purely agile approach would not have worked, and that we would have benefited from being more disciplined and putting a bit more effort into planning the work and actually following those plans.

10.2 Customisation and working with FOSS

As illustrated by the previous paragraph discussing the issues surrounding the mobile apps, there were some modifications made to DHIS2 during this project, both on the mobile apps and on the core DHIS2 application running on the server. It is not always the case that this would have been possible, depending on the software in question. If one remembers an earlier discussion we had about the potential for customisation of a software system, customisation of FOSS is always possible, given that one has the skill and capacity to do so, while customisation of proprietary software depends on what the software vendor allows you to do. In both cases, if you are an important enough customer or client, or your suggestion for changes or additions to the code are sufficiently interesting, there is the chance the vendor will make the changes for you relieving you of the cost to do so. HISP found the iCCM study interesting enough that they let one of their developers work together with us to make the necessary changes, so it would be useful to view their decision in light of the three kinds of customers/users we identified earlier, namely the *Strategic*, *Consultative* and *Transactional* customers and users.

Based on my experiences working with them, and the relationship they seem to have with HISP UiO, I feel it safe to say that Akros is definitely not just Transactional user, I would judge them more to be a Strategic user considering that HISP UiO has taken special interest in the work that Akros is doing and the novel use cases they have for DHIS2. It was not ZCAHRD or Akros that contacted HISP UiO for assistance with the iCCM study, it was UiO who contacted Akros to see if they were interested in working with ZCAHRD on this. From the start, we therefore had certain advantages that other, similar, projects probably would not have. The best example of this is perhaps the developer we worked with during this project, who was employed by HISP Vietnam and thereby paid by HISP to assist Akros with the neccessary modifications to the mobile apps and the DHIS2 core. Others might not be so lucky and would likely have to pay thousands of dollars to get the same work done, if they could find a developer already familiar with DHIS2. As the Akros HMIS adviser stated it, iCCM is "the new hot thing in development work" and he told me that during the DHIS2 Expert Academy in Oslo in mid-August, there was considerable interest in the iCCM study and the role played by DHIS2, both from donors, aid agencies and other countries. Taking this into consideration, it is easy to draw the conclusion that the iCCM study was not just "yet another project using DHIS2", but a project breaking new ground and expanding the potential user base for DHIS2 and that both parties therefore benefited from. ZCAHRD got their mHealth intervention, Akros gained a foothold in iCCM work (and have started work on their own iCCM project) and HISP got DHIS2 adopted as the software of choice in the pilot phase of a field that will only expand in the years to come.

To answer my question of whether FOSS provides any benefit when working in a resource constrained environment, I would answer that yes, it can. I am not familiar with the cost of any proprietary system that could provide the same functionalities that DHIS2 provides, but considering the budget ZCAHRD had for the implementation work, I very much doubt it. Even if funding was plentiful and software licenses could be purchased it is not given that necessary customisations could be done.

10.2.1 Working with DHIS2

As I just stated, during this project, we were fortunate enough to have the help of a developer employed by HISP to do the necessary changes to the apps and core software, but since DHIS2 is FOSS the same work that he did could easily have been done by other developers given that they were familiar with DHIS2 development. Had ZCAHRD decided to go for commercial or proprietary software, they might have needed to use multiple different systems in order to fulfill all their requirements. Indeed, their original intent was to use two different systems, cStock for R&R purposes and mTrack for patient registration and tracking, but they were judged *"not as open and accessible as DHIS2"* and that *"if we had settled for mTrack and cStock, it would be difficult to scale it to a nation-wide system."* This project could be seen as a good example of some of the benefits that FOSS provides, namely the opportunity to modify the software to fit your own specific needs instead of having to rely completely on the specific set of features it came delivered with. But again, we had the benefit of working with a developer from HISP and Akros had a "direct line" to HISP UiO since they were the ones who recommended ZCAHRD to work with Akros on this. The experience might well be very different for an organisation who decides to use DHIS2 without having any prior relationship with the HISP organisation.

If we look back to the discussion about FOSS vs proprietary software, the concept of vendor lock-in was presented, which is the more or less conscious effort of software providers to lock their customers inside their own ecosystem, storing data in proprietary formats or making it difficult to export data in order to use it in other applications. With that in mind, it should be noted that there was recently a conference in Lusaka where different organisations and agencies presented their solutions and software. Representatives from the government focused a lot on how hard it would be to integrate the presented systems with existing national systems. This could point towards the government having bad experiences with incompatible systems, or that they just understand that it is better to have systems that can freely exchange data between themselves. With regards to the data collected, DHIS2 stores it as plain text in a PostgreSQL, MySQL or H2 database, all three of which are open source and familiar to many developers. This allows the data to be imported into any application that can communicate with the chosen database, thereby reducing the risk of having ”data silos” where data is tucked away inside one system and made unavailable to all others.

When discussing FOSS in an earlier chapter, I mentioned the idea of communities versus companies. While some FOSS products have a commercial entity standing behind them, providing things like paid support, other times a user has to rely on the community around the product. DHIS2 has quite a large and active community, made evident by the number of countries and organisations using it and the traffic volume on their mailing lists, but the core development team seems a bit small. HISP focuses on the concept of ”Networks of Action” and empowering the different nodes in the HISP network to help and support each other instead of relying on HISP UiO, and while this seems to be the case when it comes to *using* DHIS2, I am left wondering if the situation is different when it comes to *developing* DHIS2. Both during the Academy and my stay with Akros, I experienced some concerns about the tight control HISP UiO keeps over DHIS2. Developers from countries like Rwanda and South Africa mentioned incidents of having attempted to submit a patch or new feature to DHIS2 core, only to have it rejected with little or no explanation. A woman from Kenya expressed her annoyance with a change to the user interface that took place from one release of DHIS2 to the next, stating that they had needed to ”spend thousands of dollars and countless hours retraining hundreds of users” because of this. This remark was more or less shrugged off by the HISP UiO representative. Akros said they could easily sympathise with her and showed me exactly what the problem was, namely a change from a purely text-based menu in one release, to a combination of icons and text in the other.The end-users are not always experienced computer users and do not have lots of experience in teaching themselves how to use a piece of software. According to Akros' own experience many users ”... don't *learn* how to use the software, they *memorize it*. So if you reorganize a menu, move a button or change the wording they can get stuck since what they memorized no longer applies.” During discussions with two employees of HISP UiO, they expressed concerns about HISP UiO having a too strong *academic focus*, and still ”treating it as a research project and not fully realising that it is now at a point where peoples lives depend on it.”

10.3 A Resource-constrained Environment

10.3.1 Trade offs

As discussed in the chapter about mobile technologies, there are always certain trade offs to be made when choosing one technology over the other. Laptop computers have more processing power and are easier to work on than any mobile phone, but mobiles are more portable, always connected and are often cheaper and more robust. Every device and technology selected has its own set of advantages and disadvantages so it is rarely possible to be completely satisfied when forced to choose and it is important to choose wisely since the technology chosen can have great impact on the possible success of a project. Neither Akros nor me had any say in the specific technologies selected for the iCCM project. ZCAHRD had already decided that the CHWs and clinic staff would use feature phones to report data and receive feedback. Considering the rural environment selected for the intervention, the amount of data to be reported, the purchase price of alternatives such as smart phones, no one disagreed with the decision that had been made. Feature phones have many desirable characteristics for usage scenario that is iCCM. First off, they are cheap and sturdy so are less likely to be stolen or broken than smart phones or computers. They use very little power and therefore have a long battery life, and while they support data transfers and have basic web browsing capabilities they do not make it as easy to use large amounts of data bundles as a smart phone would do. This is important since the CHWs will be given a certain amount of talk time and data traffic to use each month, which they will need to submit the reports. This lack of advanced functionality is of course also a downside with using feature phones, since the clinic staff will not be able to use the phones to view the reports generated by DHIS2 but will have to be sent custom forms designed for the small screen size when reviewing commodity levels and disease statistics.

There are also other trade offs to be made, which are not always so obvious at first. A specific example from this study, is the lack of support for encryption in most feature phones. During a Skype call with our developer, he made us aware of the fact that the mobile apps does not support HTTPS, only plain HTTP. We will not dig too deep into the technicalitites here, but in short HTTP stands for Hyper Text Transfer Protocol, and is the data exchange protocol the world wide web is based upon. Every website you visit uses HTTP to send you the text and images you see on your computer screen and it is also widely used as a data transfer protocol in web applications such as DHIS2. The problem with plain HTTP, is that everything is sent as plaintext meaning that everyone who is able to intercept the data traffic will be able to read its contents. This is never a good thing, but in some cases it is worse than in others, especially if the data you are transmitting is sensitive data in any way, e.g. personal health data or financial data. In order to ensure the confidentiality and integrity of the data communication, a web server can be configured to use HTTPS instead, which will encrypt all the data traffic between client and server. The problem with HTTPS is that it is more computationally expensive than just plain HTTP since everything has to be encrypted/decrypted, and the limited hardware in feature phones is not equipped to do this kind of work and therefore the DHIS2 mobile apps uses plain HTTP only. While considered non-optimal to have unencrypted data transport between the phones and server, it was rather quickly decided to be a non-issue because ”if the choice is between providing proper health care with a slight risk of loss of confidentiality, and providing poor or no health care, then it is really not a choice.”

The possible risk of loss of confidentiality of patients is also raised by ZCAHRD in their pre-study, and they conclude that the *”... risk is no higher than normal risk presented through referral records kept by CHWs.”* Though I have no evidence to back up the claim, I would assume that in rural Zambia data stored in paper records is more at risk of being stolen or read by a third party than unencrypted traffic on the cellular network.

10.3.2 Phone sourcing

Akros have extensive experience using mobile phones with DHIS2, so they were tasked with finding a suitable phone for use in the iCCM study. While Akros have previously experimented with using low-end smart phones, they have moved away from this approach and settled on using feature phones instead, though the specific model they use in their CLTS-programme was deemed too expensive for this project.

The challenge in finding a suitable phone is illustrative of how widespread smart phones have become

In order to run the DHIS2 mobile application, a mobile phone needs to have certain specifications: it needs to support Java, allow the user to install Java-based applications and have enough internal storage and memory to support the DHIS2 application. Finding out if a phone supports installation of Java applications is relatively easy, but from our developer in Vietnam we had bee quoted the very specific requirement of a heap size of at least 512kB which is not something you can just ask your average phone store sales-person about, so the only solution was to go out and buy a small selection of phones and test the application on each one.

We discussed and explored the possibility for choosing a phone with a full QWERTY-keyboard instead of just a regular numpad to make data entry easier, but they turned out to be too expensive and there were concerns about CHWs needing extra training to learn how to use them properly when they are used to regular numpads. Besides, most data fields is numbers anyway, it is only in the patient referral tracker and mentorship event report that text has to be entered so it may be that such a phone would have the opposite of the intended effect, and make data entry more cumbersome.

Finding a good feature phone is not as easy as it used to be. There seems to be a sharp divide in the phone market between relatively expensive touch-screen smartphones on the one end, and very cheap dumb phones on the other. Since this would be a bulk-purchase when a suitable phone was found, it could have been an option to approach the networks and try and negotiate a deal with them but considering the short amount of time available at this stage it was decided that is was not really serious option for the time being. Also taking into account that not all parts of Zambia has coverage from all three networks, it was decided that the best solution was to get unlocked, carrier-agnostic phones so that for every district one could go with the network providing the best coverage. If this is to be scaled nation-wide there most likely will be some talks with the network providers, considering the cost and logistics involved in providing thousands of CHWs with talk-time every month. For the purposes of the study, phones were bought from an independent store and the CHWs were provided with prepaid SIM-cards which will get refilled with a low-tier bundle every month.

During the Academy, there was one participant who voiced his annoyance at the somewhat surprising fact that the the browser based client does not support offline storage of data. Apparently this caused some frustration in some ongoing project in Zimbabwe since internet connectivity was not always present in the locations they worked in. Now, the DHIS2 website informs users that offline storage is not available in the browser based client so it should not really come as a surprise. But still, it is interesting that the least connected device, with the greatest storage capabilities does not support it, while the most connected devices, with limited storage capabilities do. For anyone unfamiliar with DHIS2, it would perhaps be natural to assume the opposite. It just goes to show that when selecting technologies, one should always strive to learn as much as possible about the different capabilities and limitations of each option before making a choice.

10.3.3 Infrastructure and Local Capacity

In the literature review chapter, I mentioned some examples of problems that are considered common when working with ICT in DCs and if not dealt with properly can lead to failure. One set of relatively common problems is poor and lacking infrastructure, especially when working in rural areas. If coming from a developed country one might not have experience with the weather causing any major problems for the electrical supply, one is used to having all kinds of spare parts easily available and may not put any thought into backup solutions in case the internet connection disappears for days on end. In order for an ICT4D project to succeed it is important to be aware of the type of problems that might occur, and plan accordingly. Although, planning for the unexpected will always prove problematic. In my case, we ran into some problems in urban areas during the initial configuration phase. Zambia is heavily dependent on hydroelectric power, getting most of their power from the Kariba dam which they share with Zimbabwe, and this year there was less rainfall than usual so low water levels led to shortages in the power supply. The consequences of this was that the entire country was subject to periodical controlled power cuts which could occur at any time of the day. There was an official schedule published for every part of Zambia and Lusaka that detailed when one could expect to loose power, but this was not always followed so it was difficult to plan ahead when one day you could have power all day long, and the next it would be gone for eight hours and then suddenly disappear again for a few hours in the evening. To make matters worse, the part of Lusaka where the Akros office is located would most commonly loose power from 0830 to 1630, basically the entire work day. Neither the Akros office nor the ZCAHRD office had a working generator until the last few days of my stay, so we were often left without both power and internet. One could of course continue working for a few hours if ones laptop was fully charged when the power went out, but since the internet went out together with the power it became difficult to work on the DHIS2 instance itself. Mobile data was still working, and it was usually better than the wired internet, but neither Akros nor ZCAHRD staff was to keen on using too much of their allotted talk time on data bundles.

The quality of the internet connection presented its own problems at times, the most common problem being having to wait for relatively long periods of time for the different parts of the DHIS2 user interface to load properly. Page loads could often take a minute or more, and would sometimes be reported as completed by the web browser but with important data and interface elements missing so reloading the page would be required to have it display properly. These two factors, poor or no internet connection as well as power outages greatly impacted our work since we were dependent on access to the DHIS2 server. The problems this caused might have been reduced somewhat by setting up a local server running a DHIS2 instance, but that would only solve the connection issue and not the lack of electricity.

In light of this, we can view the decision by Akros to employ the services of a dedicated hosting provider to run and maintain the DHIS2 instances, both their own and the one for iCCM, as a defensive measure against possible problems with the local infrastructure. Having DHIS2 running on a server outside of Zambia, in this case the USA, made the server itself immune to the effects of the frequent power cuts we experienced in Zambia. Of course, it became difficult to access the server at times, like the times the Akros office lost their internet connection, but the server itself never went offline and remained accessible. Using the services of a company with extensive experience in server administration and DHIS2 hosting, also removes the cost associated with training staff to maintain a DHIS2 instance and most likely reduces the risk of data loss or data corruption. It is not without its downsides though.

10.3.4 Building Local Capacity

Akros, as I have already mentioned, has a lack of technically skilled personnel when it comes to ICT and while that has certainly not stopped them from doing very interesting work using a selection of different technologies it does disadvantage them somewhat. They certainly have smart and talented people who are able to teach themselves how to use complex software systems such as DHIS2, but they do not have anyone who can identify areas where they could benefit from tools that are readily available and being able to develop solutions to problems that may occur. I myself saw straight away that being dependant on internet access during district DHIS2 trainings led to difficulties when the internet connection was slow and unreliable, and when I asked why they did not run a DHIS2 server on their laptop so that they could do without an internet connection, they did not even know that it was possible to do such a thing.

Because of this lack of technical skill they rely heavily on off-site consultants and developers to deal with the technical side of their operations. For a small NGO such as Akros, it would probably not be cost-effective to have an in-house developer, but having to rely on the support team of a hosting company and an external consultant to deal with any issues that might arise with their multiple DHIS2 instances leads to extra expenses that can be hard to budget for and as well as delays and loss of time since they can not always be expected to be reached when needed Just having someone in the organisation with the skills to do basic server administration would in my opinion be of great benefit for them. To illustrate, at one point their CLTS DHIS2 instance started malfunctioning and refused connections from mobile phones leading to half the office scrambling to figure out what was going on. They tried getting in touch with the hosting company, who they only have email and Skype contact details for, without any luck because of timezone differences, but they got in touch with their regular consultant who was able to set things right again after a few hours of work. During this, I was the only one in the Akros office with access to the server itself, and not just the DHIS2 web interface, since I had been given root access to the server, by the hosting company, some days prior in order to retrieve up-to-date backups of the CLTS database. Had I, or anyone else in the office, had any experience with administrating a DHIS2 installation, we might have been able to resolve the issue without having to enlist the help of a consultant who, in Akros own words, is *"fantastic at what he* *does, but he's like* really *expensive, we pay him a lot."*

Building local capacity is something HISP is very focused on and can be of great importance to the eventual success or failure of an ICT4D project. Members of ZCAHRD staff was given thorough training in the use and configuration of DHIS2 for their specific use case, and their iCCM Program Manager was very involved and interested in "learning the ropes" from the start and has great personal investment in the project. Successful information system deployments often depend on, amongst other things, the enthusiasm and knowledge about the system amongst members of the organisation deploying the system. Some have coined the term "local champions" for this type of user, and in my opinion the ZCAHRD Program Manager fits the role of such a champion and without his great personal involvement the project would probably have a lower probability of success. What they have not received, is any training in the more technical aspects of DHIS2. Not only is this because of time restraints and that such training would have been beyond the scope of this project, but there is no one in Akros who have the technical skill and expertise necessary to provide such training if it had been requested. In other words, there is in this project a severe lack of local capacity to handle any future technical problems that might arise, with ZCAHRD being dependant upon Akros and Akros depending on consultants and developers located in other countries. The downside to this, together with the use of a hosting company that abstracts away all the details concerning data backups, stable and uninterrupted power supply and internet connection, the need for skilled staff to maintain the installation and so on, is that by offshoring all this work no contribution is made to the local *learning* economy, as discussed previously, as well as the economy as a whole. The danger is that they become dependent on a system they cannot maintain by themselves. On the other hand, ZCAHRD is only an intermediary and it is intended that if this study is successful and shows promising results, it will be handed over to the MoH who will then take the responsibility for rolling it out to the entire country. The MoH already uses DHIS2 to some degree, and have more skill and resources to draw upon than ZCAHRD so they will hopefully have the capacity needed to maintain and expand upon the system we built.

10.3.5 Funding problems

Many of the problems that was encountered during this project can be said to be the result of the overarching problem with funding. More specifically, it was not a *lack* of funds that caused problems, though that was certainly a part of it, but rather the *administration* of the available funds. the constant need to wait for funding and payments to go through. As I mentioned earlier in this chapter, Akros, and thereby myself, had to wait for approximately two weeks after accepting the contract before the payment details was worked out, so work was delayed already from the start. This was caused by Akros wanting to be paid in US dollars, while ZCAHRD had gotten their funding from UNICEF in Zambian Kwacha which had weakened quite a bit against the dollar during the time since they received the funds. The two week delay worsened the problem since during that time the kwacha was further weakened. While Akros was ready to start work, they did not want to dedicate any resources to it before ZCAHRD agreed to the terms of payment. So if the payment issue had been resolved earlier, we could have started work two weeks earlier, which would probably have had a considerable impact considering that this would have doubled the time the Akros HMIS adviser would have had available to work on the project on-site.

Further, after Akros' work was done and "all" that remained to do was train the CHWs, the trainings were delayed for several weeks and again this was directly attributed to the funding. The DHIS2 instance was up and running and fully configured, the training materials had been created and mobile phones for the CHWs and HC staff procured, but the transfer of funds from UNICEF to ZCAHRD was delayed because of some problems with the paperwork, and this led to the trainings being delayed. When considering the long duration of the study, which is to last until June 2016, a delay of two weeks should likely not have too great of an impact on the final results.

Akros shared my impression that the process had been a bit disorganised and that many issues could have been avoided, or solved in a more timely manner, had we spent more time on planning beforehand. While I initially got the impression that projects like these were something Akros had experience with, I was now told that this was in face the first time their HMIS advisor had worked on a project where both the budget and deadline was so tight with a set fee and his previous experience was with projects without a strict deadline and budget which gave them more leeway in how they approached them. I must stress that my impression was based on what I experienced then and there. It is not like they lied to me and said "oh, we got lots of experience with this".

10.3.6 Recommendations

ICT projects often provide new and innovative solutions that require new ways of working and new skills to be acquired. Depending on how the responsibility for maintaining, and perhaps extending, a project is delegated it may lead to local industries getting a windfall. In this case, mobile phones and data bundle subscriptions had to be purchased and if scaled to the entire country that will mean thousands of phones and subscriptions which can perhaps boost the local economy. The MoH might want to take full control of the operations in a nationwide rollout, and employ and train staff in order to gain the technical capacity to maintain DHIS2 server installations and gain expertise in that area, contributing to increasing the skilled labour pool.

I would recommend Akros to take more time in the start of any project to identify the risks and challenges involved and choose an appropriate approach. Far too many problems encountered during this project was due to poor planning in the beginning. With a project such as this, with a fixed budget and deadline and specifications already worked out, an approach based on the waterfall model might be best suited.

Distributed development works, as we can tell from the multitude of successful software projects that rely on it, but perhaps it works better for development of software and not as good for configuration of software. When developing software, one has the benefit that the source code is just text and design documents and specifications are either text and/or images, both of which are easy to share and where it easy to point out and see the changes that have been made from one version to the next. That is not always the case when working on configuring a system, especially when said system is configured through a graphical interface and there exist no immediate visual proof of what changes have been made. Even though the same instance of the system is available to everyone, the team members need to communicate to each other what configuration changes they have made to make sure everyone is on the same page and that nothing is left out. Otherwise one has to go through the entire system looking for changes since they do not stand out. Something to keep in mind is that whether it is development or configuration, it is more difficult to succeed in working distributed than when all members of the team is co-located, so care should be taken to ensure that the work is organised in such a way that it is actually possible to work in a distributed manner. And if team members are located in different time zones, it is highly beneficial that there is at least some overlap between working hours otherwise it can introduce unnecessary delays.

If I should make some suggestions for the further development and work on DHIS2, I would point out that there is a lack of easily available training materials for new users of DHIS2, something that became evident both during the DHIS2 Academy as well as during my work on the iCCM project.

Chapter 11

Conclusion

As we have seen, knowing what challenges and problems one might face when working in an unfamiliar environment is important, and precautions should be taken to guard against those problems it is possible to avoid. There are not only technical requirements that needs to be met for a project to be successful, many other requirements has to fulfilled that can be just as, or more important, than the technical ones. Funding is critical to any development project, since if there is no funding, there is no money to pay for necessary equipment and manpower. But not only does one have to secure funding, they also need to be administered correctly and efforts should be taken to ensure that payments happen on time to avoid delays. Depending on the severity of such delays, they may impede the whole process and lead to further problems down the road. Understanding what motivates people who will be part of the project, especially if they are volunteers such as the CHWs in Zambia, can be critical for a projects success since without their participation it does not matter how good the technical solution is. Small details, such as the number of fields in a form, or the transition from text-based menus to icons can have a surprsingly great impact so enlisting the help of someone who has done similar work before can greatly impact the results in a project.

11.1 Limitations

Optimally, I would have gone further down the organisational hierarchy and gotten some time in the field while in Zambia in order to study how the CHWs and clinic staff actually carried out their iCCM work with the paper based system that was currently in place. It would most likely have given me a better understanding of the challenges involved and This did not happen, for multiple reasons. Foremost among them, the fact that I could have done more to explore the possibilities for fieldtrips of that kind, but I was there as a student who had only I was presented with the opportunity to join Akros on a fieldtrip, but that wo

It was my hope that I would get some weeks worth of data from the iCCM study. iCCM is a relatively new field, and since this is a pilot study and the first time mHealth solutions are used for iCCM it will most likely yield plenty of interesting data in the coming months. One unanswered question I am left with, is how much reporting rates will suffer, if at all, from the size of the forms CHWs are supposed to submit.

Considering that action research is based on an iterative process where one diagnoses the problem situation, plans appropriate actions, implement those actions and then evaluate and reflect upon the result, the limitations just mentioned means that I did not get to follow all the steps. Most of what I did was in the implementation phase, with some diagnosis and planning being done but much of it already done before I got involved. Since the results of the intervention is not yet available, the evaluation and reflection stage can only consider the previous phases in the process and not the real world results of those phases.

11.2 Further research

While distributed development has lots of tool to help it work smoothly, tools such as online code hosting, mailing lists, etc make it easy to keep track of what features others are working on and what changes have been made to the code base, the same is not the case when it comes to system configuration. When configuring DHIS2, there is to my knowledge no way of knowing what changes others have made to the system so it becomes important to tell everyone working on it what changes have been made and where. Further research could be done on the topic of distributed systems configuration.

Bibliography

Avgerou, C., 2008. Information systems in developing countries: A critical research review. Journal of Information Technology 23, 133–146.

Boehm, B. W., 1988. A spiral model of software development and enhancement. Computer 21 (5), 61–72.

Braa, J., Sahay, S., 2012. Integrated Health Information Architecture: Power to the Users: Design, Development and Use. Matrix Publishers.

Byrne, E., Jolliffe, B., 2007. Free and open source software-development as freedom. In: Proceedings of the 9th International Conference on Social Implications of Computers in Developing Countries. IFIP TC.

Davison, R., Vogel, D., Harris, R., Jones, N., 2000. Technology leapfrogging in developing countries-an inevitable luxury? The Electronic Journal of Information Systems in Developing Countries 1.

Déglise, C., Suggs, L. S., Odermatt, P., 2012a. Short message service (sms) applications for disease prevention in developing countries. Journal of medical Internet research 14 (1), e3.

Déglise, C., Suggs, L. S., Odermatt, P., 2012b. Sms for disease control in developing countries: a systematic review of mobile health applications. Journal of Telemedicine and Telecare 18 (5), 273–281.

Ernst, D., Lundvall, B.-A., 2004. Globalization, Economic Development and Inequality. Edward Elgar Publishing Limited, Ch. 9: Information technology in the learning economy: challenges for developing countries, pp. 258–287.

Fitzgerald, B., 2006. The Transformation of Open Source Software. MIS Quarterly 30 (3), 587.

Fraser, H. S., Blaya, J., 2010. Implementing medical information systems in developing countries, what works and what doesn't. In: AMIA Annual Symposium Proceedings. Vol. 2010. American Medical Informatics Association, pp. 232–236.

Freifeld, C. C., Chunara, R., Mekaru, S. R., Chan, E. H., Kass-Hout, T., Iacucci, A. A., Brownstein, J. S., 2010. Participatory epidemiology: use of mobile phones for community-based health reporting. PLoS Med 7 (12), e1000376.

Ghosh, R. A., 2003. Licence fees and gdp per capita: The case for open source in developing countries. First Monday 8 (12).

GSMA Intelligence, 2014. Measuring Mobile Penetration. https://gsmaintelligence.com/research/2014/05/measuring-mobile-penetration/430/, [Online: Last accessed 24.11.2015].

Heeks, R., 2002. Information Systems and Developing Countries: Failure, Success, and Local Improvisations. The Information Society 18 (February 2015), 101–112.

Heeks, R., February 2005. Icts and the mdgs : On the wrong track? I4Donline.Net, 9–12.

Heeks, R., 2008. ICT4D 2.0: The next phase of applying ICT for international development. Computer 41, 26–31.

HISP UiO, 2014. HISP UiO Strategy 2014-2016. http://www.mn.uio.no/ifi/english/research/networks/hisp/hisp-uio-strategy-13.03.2014-2014-2016.pdf, [Online; last accessed 24.11.2015].

Hoe, N., 2006. Breaking barriers: the potential of free and open source software for sustainable human development; a compilation of case studies from across the world.
URL http://idl-bnc.idrc.ca/dspace/bitstream/10625/37302/1/127878.pdf

Kvasny, L., Truex, D., 2001. Defining away the digital divide: A content analysis of institutional influences on popular representations of technology. In: Realigning Research and Practice in Information Systems Development. Springer, pp. 399–414.

Lakhani, K. R., Wolf, R. G., 2005. Why Hackers Do What They Do : Understanding Motivation and Effort in Free / Open Source Software Projects 1 By. Perspectives on Free and Open Source Software, 1–27.

Lucas, H., 2008. Information and communications technology for future health systems in developing countries. Social Science and Medicine 66, 2122–2132.

Marnewick, C., 2012. A longitudinal analysis of ict project success. In: Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference. ACM, pp. 326–334.

May, C., 2006. Escaping the TRIPs' trap: The political economy of free and open source software in Africa. Political Studies 54 (1), 123–146.

Penicaud, Claire and Katakam, Arunjay, 2013. State of the Industry 2013 Mobile Financial Services for the Unbanked. http://www.gsma.com/mobilefordevelopment/wp-content/uploads/2014/02/SOTIR_2013.pdf, [Online: Last accessed 24.11.2015].

Pew Research Center, February 2014. Emerging Nations Embrace Internet, Mobile Technology. http://www.pewglobal.org/files/2014/02/Pew-Research-Center-Global-Attitudes-Project-Technology-Report-FINAL-February-13-20147.pdf, [Online: Last accessed 24.11.2015].

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.

PWC, GSMA, 2012. Touching lives through mobile health: Assessment of the global market opportunity. http://www.pwc.in/assets/pdfs/telecom/gsma-pwc_mhealth_report.pdf, [Online: Last accessed 24.11.2015].

Sanner, T. A., Roland, L. K., Braa, K., 2012. From pilot to scale: Towards an mHealth typology for low-resource contexts. Health Policy and Technology 1 (3), 155–164. URL http://dx.doi.org/10.1016/j.hlpt.2012.07.009

Seebregts, C. J., Mamlin, B. W., Biondich, P. G., Fraser, H. S. F., Wolfe, B. A., Jazayeri, D., Miranda, J., Blaya, J., Sinha, C., Bailey, C. T., Kanter, A. S., 2010. Human Factors for Capacity Building. Lessons learned from the OpenMRS Implementers Network. Yearbook of medical informatics, 13–20.

Sein, M., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R., 2011. Action design research.

Sen, A., 1999. Development as Freedom. Oxford University Press.

Susman, G. I., Evered, R. D., 1978. An assessment of the scientific merits of action research. Administrative science quarterly, 582–603.

United Nations, 2013. The Millenium Development Goals Report 2013. http://www.un.org/millenniumgoals/pdf/report-2013/mdg-report-2013-english.pdf, [Online: Last accessed 24.11.2015].

Wade, R. H., 2002. Bridging the digital divide: New route to development or new form of dependency? Global Governance 8 (4), 443–466.

Walsham, G., Sahay, S., 2006. Research on information systems in developing countries: Current landscape and future prospects. Information Technology for Development 12 (February 2015), 7–24.

Wazny, K., Sadruddin, S., Zipursky, A., Hamer, D. H., Jacobs, T., Kallander, K., Pagnoni, F., Peterson, S., Qazi, S., Raharison, S., et al., 2014. Setting global research priorities for integrated community case management (iccm): Results from a chnri (child health and nutrition research initiative) exercise. Journal of global health 4 (2).

West, M., Chew, H. E., 2014. Reading in the mobile era: A study of mobile reading in developing countries. http://unesdoc.unesco.org/images/0022/002274/227436e.pdf, [Online: Last accessed 24.11.2015].

Zambia Central Statistical Office, 2014. Zambia in Figures: 1964 - 2014. http://www.zamstats.gov.zm/latest/Zambia%20in%20Figures%20Booklet.pdf, [Online: Last accessed 24.11.2015].

Appendix

The features ZCAHRD required of the DHIS2 implementation, is shown in figure 11.1.

Figure 11.2 shows the original, paper based R & R reporting form that the CHWs are using. We replicated this as best we could in DHIS2, with the result shown in figure 11.3. Each section in the DHIS2 form corresponds to a row in the matrix in the original form. The same is shown with the disease reporting forms, figure 11.4 showing the part of the paper based disease reporting form that is used to aggregate data, and figure 11.5 showing the same form in DHIS2. Again, each section in the DHIS2 form corresponds to a row in the paper form. Note that the element ”of whom NO. OF CHILDREN AGE $<$ 5” is not present in the DHIS2 form, since this can be calculated from the ”Female Under-fives” and ”Male Under-fives” elements.

In figure 11.6 I have attempted to illustrate the actors and actions involved in iCCM at the actionable level. Actors such as ZCAHRD and MoH is not present here.

A model of the workflow in R & R reporting, when no errors are found in the data, is shown in figure 11.7, and figure11.8 shows the workflow when reporting aggregated disease data. Both models are based on BPMN, but does only somewhat follow the official notation.

The remaining figures are the training materials I developed in order to help both myself and ZCAHRD staff when learning how to use and configure DHIS2. They may seem a bit confusing to those who have never used DHIS2, but I have been told they have been very helpful for Akros during their trainings. Figure 11.9 illustrates how users are given access to data sets and tracker programs based on the user role and organisation units they are assigned to. Figure 11.10 illustrates the steps needed to create a new user in DHIS2. Figure 11.11 shows the steps involved when creating a data set or tracker program in DHIS2. Figure 11.12 represents the relationship between the parts that make up data sets and tracker programs. Figure 11.13 extends upon figure 11.12 and shows the dependencies between the different parts.

Required DHIS 2 features for our iCCM study needs

1. - Weekly aggregate report from community health workers (CHWs), using feature phone and working out of a health posts, tracking commodities received during the week, used during the week, and commodities on hand.

2. - HTML Report customized by clinic (covering several health posts) showing current commodity levels by healthpost and need for resupply

3. - HTML Report customized by clinic (covering several CHWs) showing current commodity levels by each CHW and need for resupply

4. - Automated weekly SMS to clinic with HTML commodities report.
- Weekly aggregate report from CHWs on caseload (Malaria, Pneumonia and Diarrhea cases treated by each CHW)

5. - Feature phone event capture of patient referral from health post level to clinic Capture: Patient demographics, health condition, and treatment administered. (HISP Uganda Tracker app????)

6. -Feature phone event capture of patient referral from CHWs to clinic Capture: Patient demographics, health condition, and treatment administered. (HISP Uganda Tracker app????)

7. - Automated SMS notifying clinic of patient referral.

8. - Automated monthly SMS reminder to clinics to conduct mentorship

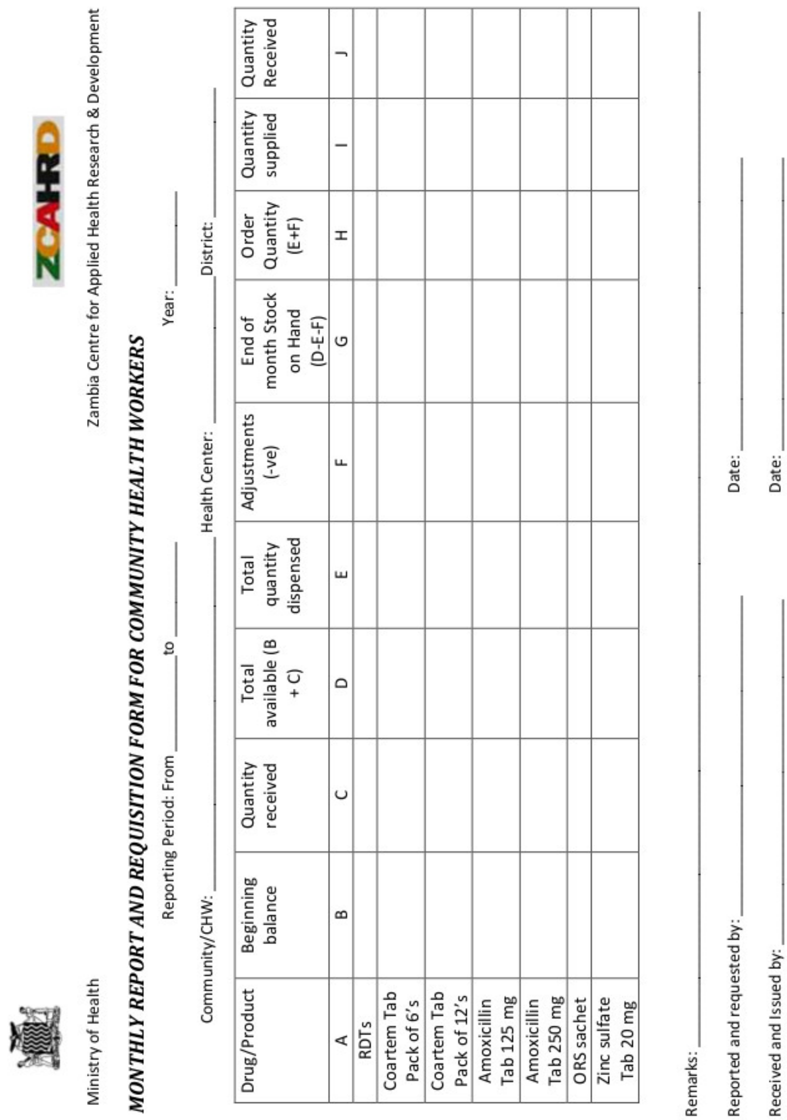
9. - Monthly aggregate report from clinic level on administering mentorship

10. - Automated monthly SMS to CHWs to reinforce appropriate use of iCCM classification and treatment algorithm

11. - Phone and app troubleshooting videos (Akros and ZCHARD will prepare)

12. - Disease specific case management instructional videos (Akros and ZCHARD will prepare)

Figure 11.1: Requirements specification



Ministry of Health

Zambia Centre for Applied Health Research & Development

MONTHLY REPORT AND REQUISITION FORM FOR COMMUNITY HEALTH WORKERS

Reporting Period: From ____________ to ____________ Year: ____________

Community/CHW: ____________ Health Center: ____________ District: ____________

Drug/Product	Beginning balance	Quantity received	Total available (B + C)	Total quantity dispensed	Adjustments (-ve)	End of month Stock on Hand (D-E-F)	Order Quantity (E+F)	Quantity supplied	Quantity Received
A	B	C	D	E	F	G	H	I	J
RDTs									
Coartem Tab Pack of 6's									
Coartem Tab Pack of 12's									
Amoxicillin Tab 125 mg									
Amoxicillin Tab 250 mg									
ORS sachet									
Zinc sulfate Tab 20 mg									

Remarks: ____________

Reported and requested by: ____________ Date: ____________

Received and Issued by: ____________ Date: ____________

Figure 11.2: Originial Paper based R& R form

Drugs - Report & Requisition

Drugs report and requisition report form submitted weekly by an iCCM trained CHWs

Organization unit identifier	
Period	2014W1 - 2013-12-30 - 2014-01-05

RDTs

RDTs (Max Balance)	
RDTs (Beginning Balance)	
RDTs (Quantity Received)	
RDTs (Quantity Dispensed)	
RDTs (Adjustments)	
RDTs (Order Quantity)	

Coartem Tab Pack of 6's

Coartem Tab Pack of 6's (Max Balance)	
Coartem Tab Pack of 6's (Beginning Balance)	
Coartem Tab Pack of 6's (Quantity Received)	
Coartem Tab Pack of 6's (Quantity Dispensed)	
Coartem Tab Pack of 6's (Adjustments)	
Coartem Tab Pack of 6's (Order Quantity)	

Coartem Tab Pack of 12's

Coartem Tab Pack of 12's (Max Balance)	
Coartem Tab Pack of 12's (Beginning Balance)	
Coartem Tab Pack of 12's (Quantity Received)	
Coartem Tab Pack of 12's (Quantity Dispensed)	
Coartem Tab Pack of 12's (Adjustments)	
Coartem Tab Pack of 12's (Order Quantity)	

Amoxicillin Tab 125 mg

Amoxicillin Tab 125 mg (Max Balance)	
Amoxicillin Tab 125 mg (Beginning Balance)	
Amoxicillin Tab 125 mg (Quantity Received)	
Amoxicillin Tab 125 mg (Quantity Dispensed)	
Amoxicillin Tab 125 mg (Adjustments)	
Amoxicillin Tab 125 mg (Order Quantity)	

Amoxicillin Tab 250 mg

Amoxicillin Tab 250 mg (Max Balance)	
Amoxicillin Tab 250 mg (Beginning Balance)	
Amoxicillin Tab 250 mg (Quantity Received)	
Amoxicillin Tab 250 mg (Quantity Dispensed)	
Amoxicillin Tab 250 mg (Adjustments)	
Amoxicillin Tab 250 mg (Order Quantity)	

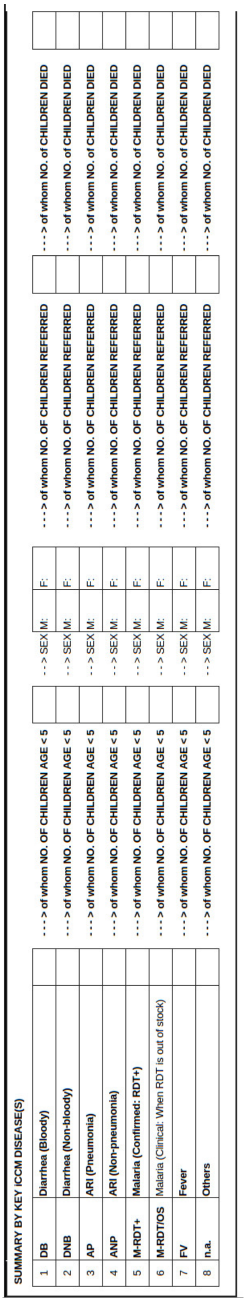
Zinc Sulfate Tab 20 mg

Zinc sulfate Tab 20 mg (Max Balance)	
Zinc sulfate Tab 20 mg (Beginning Balance)	
Zinc sulfate Tab 20 mg (Quantity Received)	
Zinc sulfate Tab 20 mg (Quantity Dispensed)	
Zinc sulfate Tab 20 mg (Adjustments)	
Zinc sulfate Tab 20 mg (Order Quantity)	

ORS sachet

ORS sachet (Max Balance)	
ORS sachet (Beginning Balance)	
ORS sachet (Quantity Received)	
ORS sachet (Quantity Dispensed)	
ORS sachet (Adjustments)	
ORS sachet (Order Quantity)	

Figure 11.3: R & R form in DHIS2



SUMMARY BY KEY iCCM DISEASE(S)

1	DB	Diarrhea (Bloody)		---> of whom NO. OF CHILDREN AGE < 5		--> SEX M:	F:	---> of whom NO. OF CHILDREN REFERRED		---> of whom NO. of CHILDREN DIED
2	DNB	Diarrhea (Non-bloody)		---> of whom NO. OF CHILDREN AGE < 5		--> SEX M:	F:	---> of whom NO. OF CHILDREN REFERRED		---> of whom NO. of CHILDREN DIED
3	AP	ARI (Pneumonia)		---> of whom NO. OF CHILDREN AGE < 5		--> SEX M:	F:	---> of whom NO. OF CHILDREN REFERRED		---> of whom NO. of CHILDREN DIED
4	ANP	ARI (Non-pneumonia)		---> of whom NO. OF CHILDREN AGE < 5		--> SEX M:	F:	---> of whom NO. OF CHILDREN REFERRED		---> of whom NO. of CHILDREN DIED
5	M-RDT+	Malaria (Confirmed: RDT+)		---> of whom NO. OF CHILDREN AGE < 5		--> SEX M:	F:	---> of whom NO. OF CHILDREN REFERRED		---> of whom NO. of CHILDREN DIED
6	M-RDT/OS	Malaria (Clinical: When RDT is out of stock)		---> of whom NO. OF CHILDREN AGE < 5		--> SEX M:	F:	---> of whom NO. OF CHILDREN REFERRED		---> of whom NO. of CHILDREN DIED
7	FV	Fever		---> of whom NO. OF CHILDREN AGE < 5		--> SEX M:	F:	---> of whom NO. OF CHILDREN REFERRED		---> of whom NO. of CHILDREN DIED
8	n.a.	Others		---> of whom NO. OF CHILDREN AGE < 5		--> SEX M:	F:	---> of whom NO. OF CHILDREN REFERRED		---> of whom NO. of CHILDREN DIED

Figure 11.4: Originial Paper based disease reporting form

iCCM Disease Managment

Organization unit identifier
Period 2014W1 - 2013-12-30 - 2014-01-05

Diarrhea (Bloody)

Diarrhea - Blood (Total Cases Seen)
Diarrhea - Blood (Female Under-fives)
Diarrhea - Blood (Male Under-fives)
Diarrhea - Blood (Referred)
Diarrhea - Blood (Died)

Diarrhea (Non-bloody)

Diarrhea - Non Blood (Total Cases Seen)
Diarrhea - Non Blood (Female Under-fives)
Diarrhea - Non Blood (Male Under-fives)
Diarrhea - Non Blood (Referred)
Diarrhea - Non Blood (Died)

ARI (Pneumonia)

Pneumonia (Total Cases Seen)
Pneumonia (Female Under-fives)
Pneumonia (Male Under-fives)
Pneumonia (Referred)
Pneumonia (Died)

ARI (Non-pneumonia)

Non - Pneumonia (Total Cases Seen)
Non - Pneumonia (Female Under-fives)
Non - Pneumonia (Male Under-fives)
Non - Pneumonia (Referred)
Non - Pneumonia (Died)

Malaria (Confirmed)

Confirmed Malaria (RDT+) (Total Cases Seen)
Confirmed Malaria (RDT+) (Female Under-fives)
Confirmed Malaria (RDT+) (Male Under-fives)
Confirmed Malaria (RDT+) (Referred)
Confirmed Malaria (RDT+) (Died)

Malaria (Clinical)

Clinical Malaria (Total Cases Seen)
Clinical Malaria (Female Under-fives)
Clinical Malaria (Male Under-fives)
Clinical Malaria (Referred)
Clinical Malaria (Died)

Fever

Fever Cases (Total Cases Seen)
Fever Cases (Female Under-fives)
Fever Cases (Male Under-fives)
Fever Cases (Referred)
Fever Cases (Died)

Newborn Care Management

Newborn (Visited)
Newborn (Referred)
Newborn (Died)

Figure 11.5: Disease reporting form in DHIS2

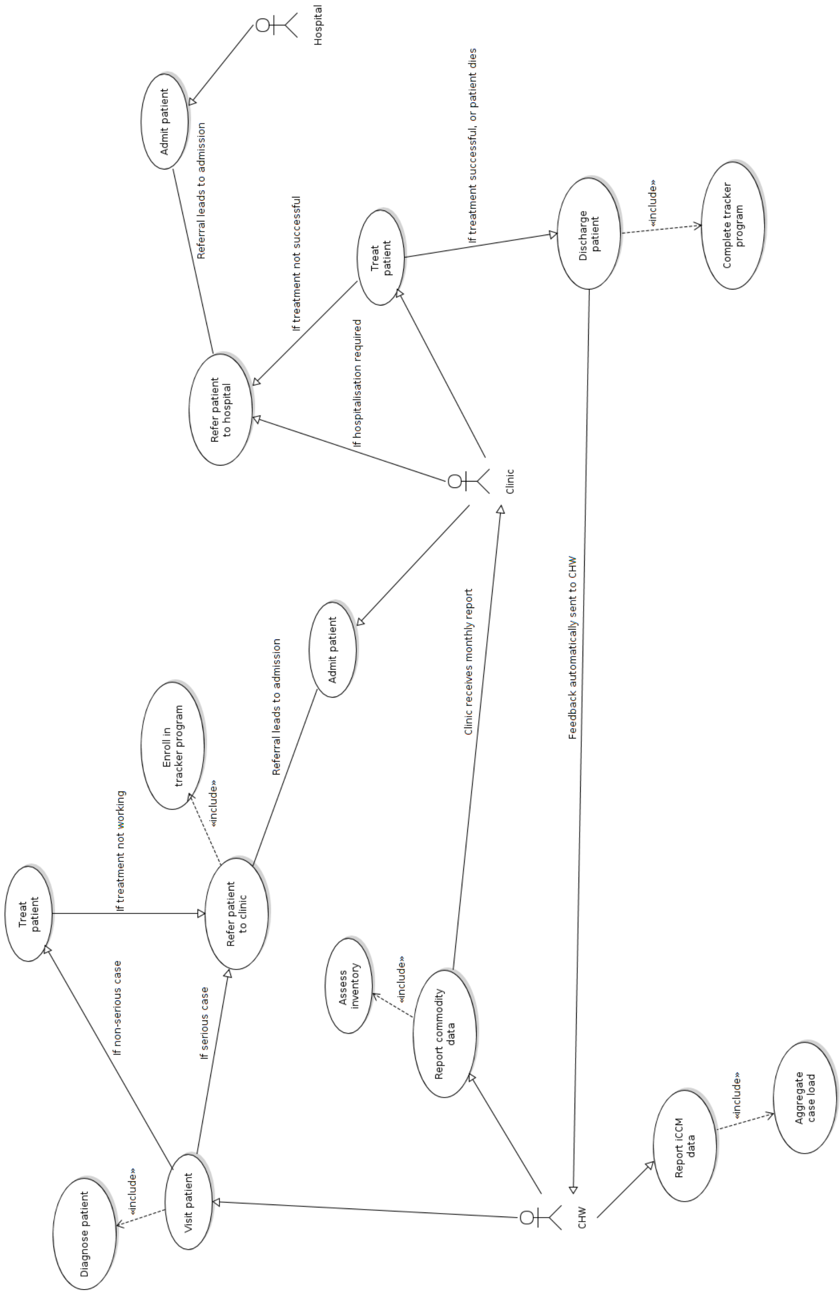


Figure 11.6: Actors and actions involved in iCCM

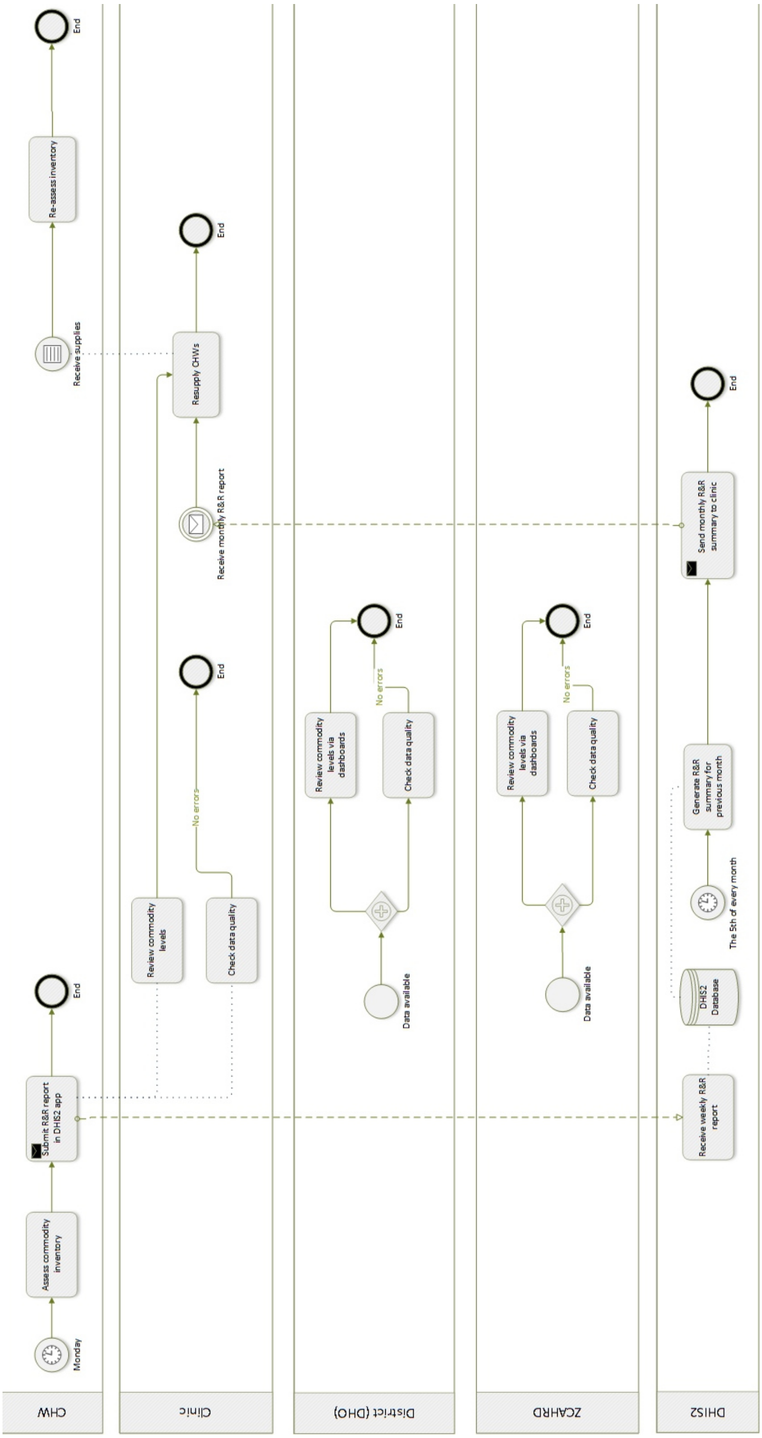


Figure 11.7: Workflow R&R reporting when no errors in data

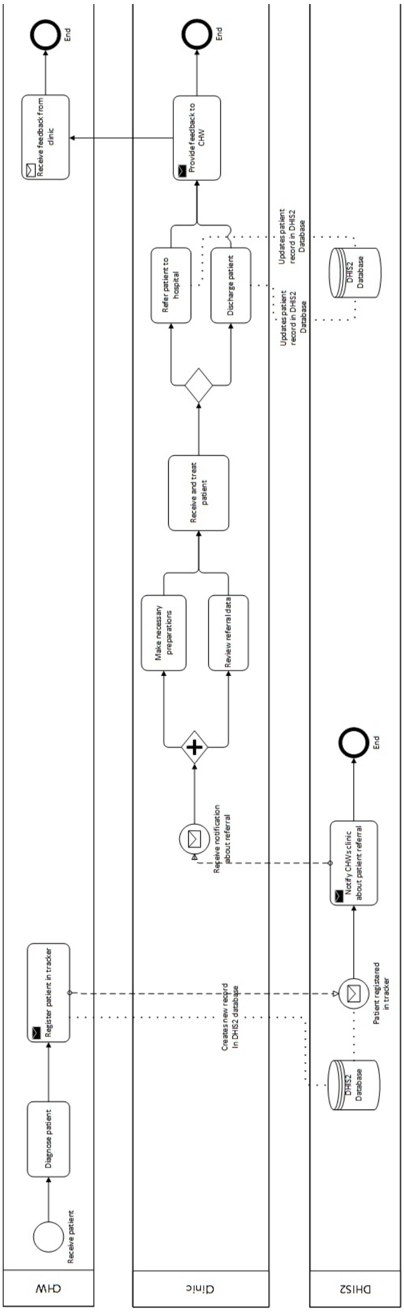


Figure 11.8: Disease Reporting; workflow

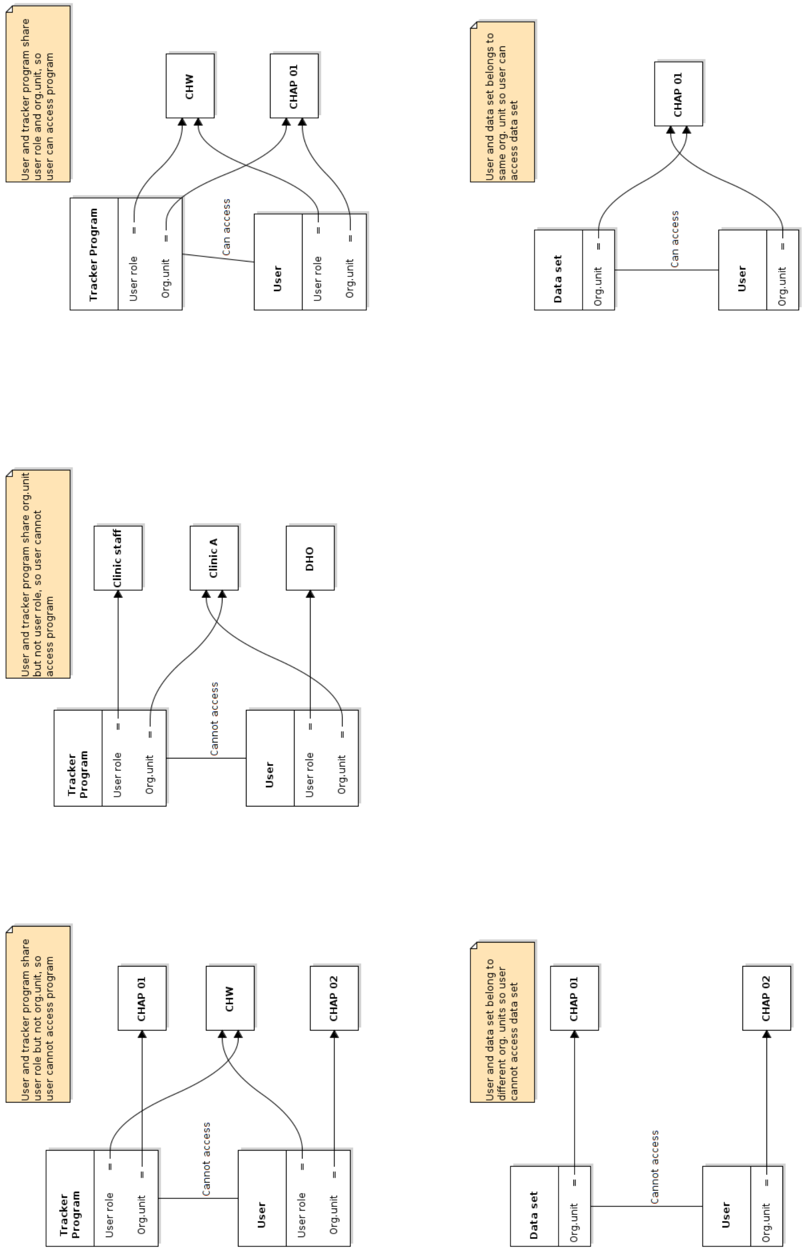


Figure 11.9: User access DHIS2

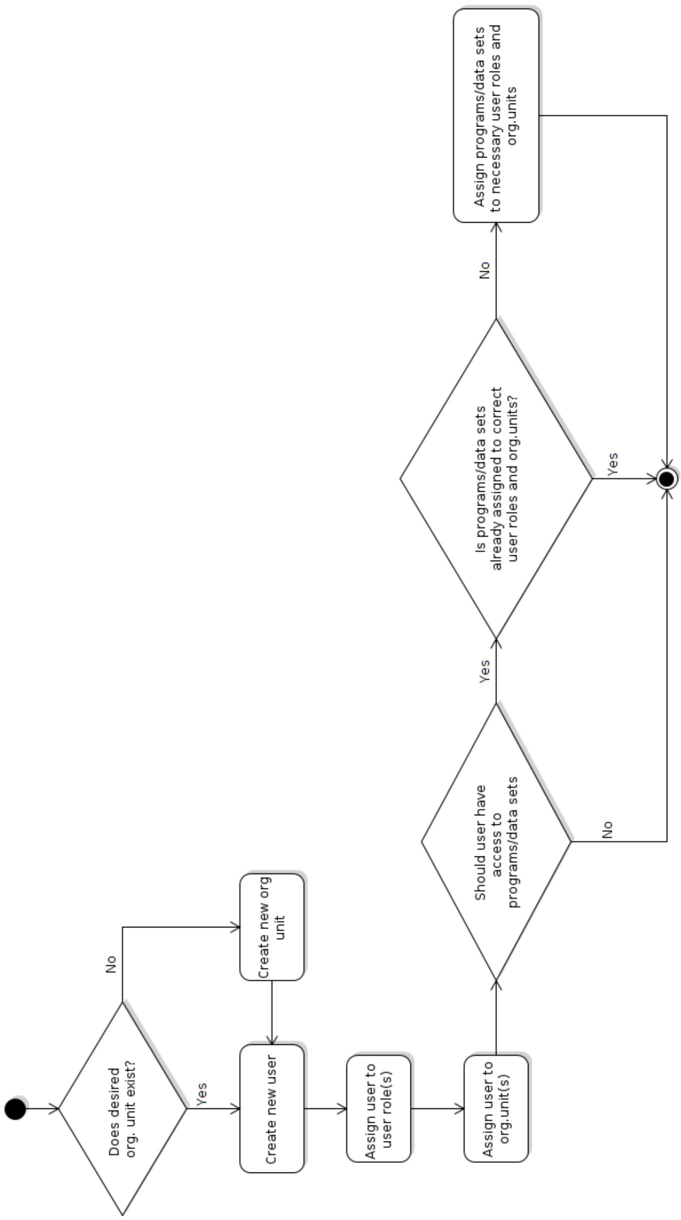


Figure 11.10: Create new user in DHIS2

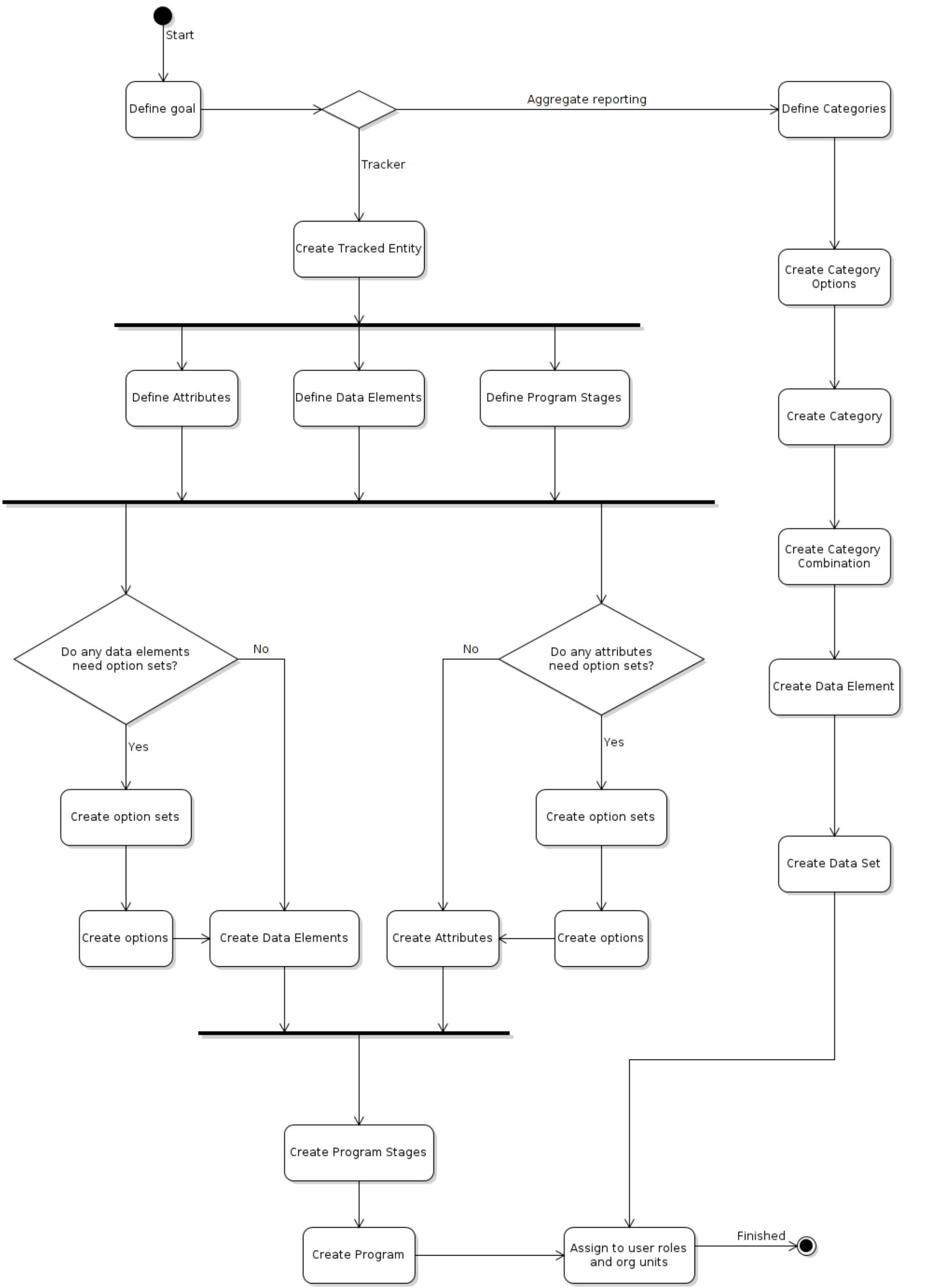


Figure 11.11: Create new Tracker Program or Dataset in DHIS2

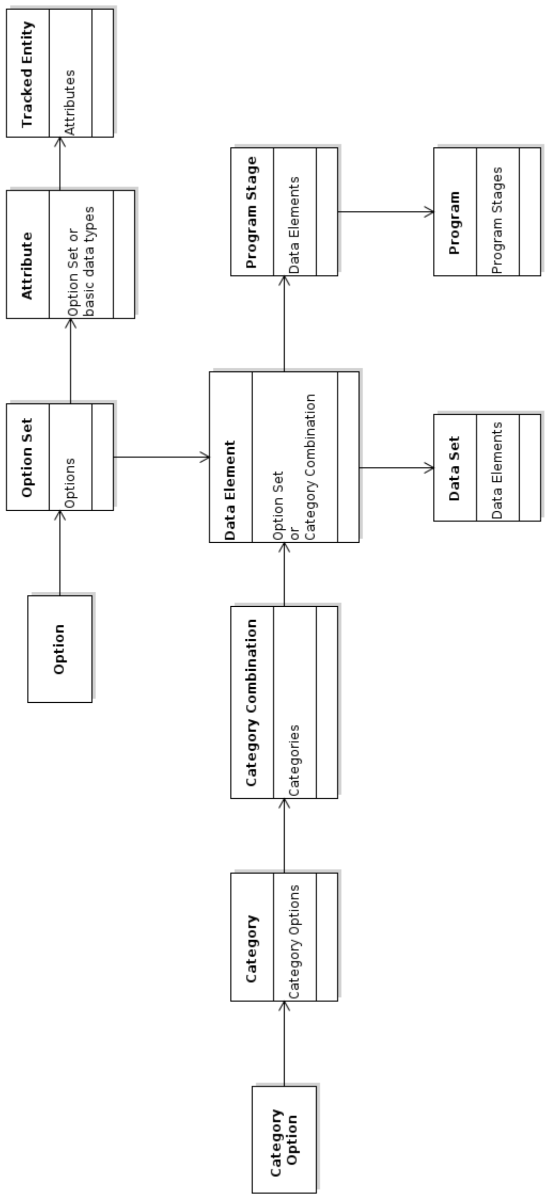


Figure 11.12: Data elements in DHIS2

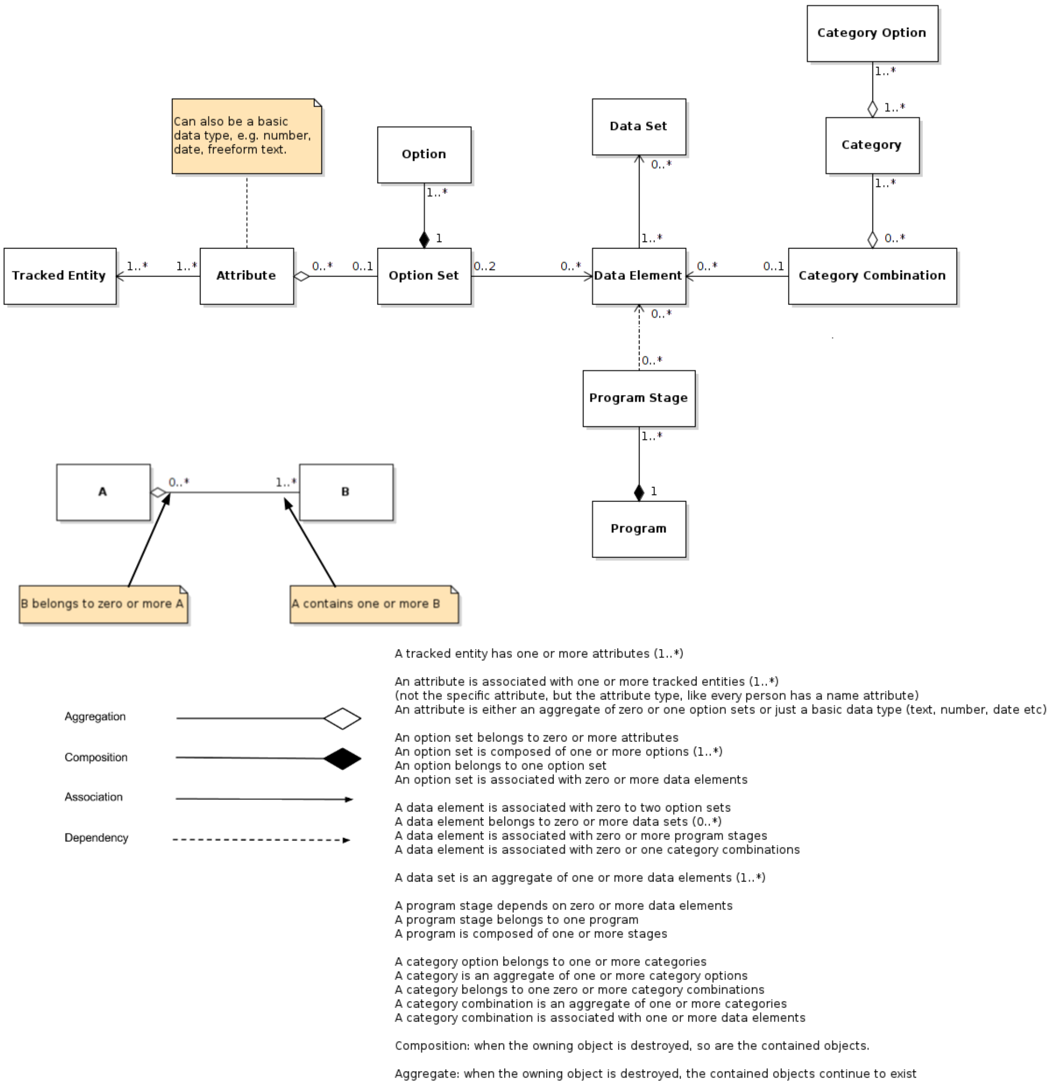


Figure 11.13: Dependencies between the different parts of a DHIS2 data set or tracker program