The London School of Economics and Political Science

When local routines meet global technology: A case study on the role of context in application development in Kampala

Kari Matti Koskinen

A thesis submitted to the Department of Management, Information Systems and Innovation Group, of the London School of Economics and Political Science for the degree of Doctor of Philosophy, London, October 2017

Declaration

I certify that the thesis I have presented for examination for the PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

I declare that my thesis consists of 92.022 words.

Abstract

The growth in mobile ownership and increase in Internet connectivity has led many developing countries to actively pursue the creation of application development sectors within their economies. Application development appears as a feasible option to even the less developed regions, as most of the technological resources needed in application development can be accessed with relative ease and low cost, no matter the location. What is more, in addition to economic benefits the applications can also have a role in solving societal challenges. Although technologically application development seems relatively straightforward, what remains less well understood is how contextual factors, such as norms and cultures, impact the application development process within these countries.

This thesis approaches the research area through the concept of societal routines. Societal routines are seen as proxies to local context, and the aim of this research is to analyse how these routines affect application development throughout the application development process. The research takes the form of a case study and studies the topic from the perspective of application development that occurs in Kampala, Uganda. Although the local developers and start-ups are generally comfortable with the technological affordances provided by the technological resources, the local context poses certain conditions, which not only impact how applications are built, but also what kinds of application are developed.

The results show how existing societal routines form the basis for the applications, and how technology that originates from outside carries meanings and structures that may or may not fit with local realities. Overall, the research proposes a framework for understanding context and its impact on application development. With this, it aims to contribute to our understanding on local technology production, technology implementation and digital divide in developing countries. Furthermore, it also questions the role often given to technology in addressing societal challenges.

Acknowledgments

First of all, I wish to thank my supervisor, Shirin Madon, for her continuous support and guidance throughout my thesis, as there were times when I felt less than confident of my abilities to write a thesis not to mention to be able to submit one. I do not think I have met another person with such patience and wisdom, and even when I was not quite sure what to do, she always managed to put me back on the right track and made me believe in myself, and all this while allowing me to do the research I wanted. No matter what the outcome of this thesis will be, I can already say that without her it would not be even close to the level that it is now.

I would also like to thank the faculty and staff at my department, who have helped me in various ways throughout this journey by giving me feedback, advice and the chance of continuously learning more and developing my skills. Special thanks go to Nathalie Mitev, Maha Shaikh and my second supervisor Carsten Sørensen, as well as to Leo Beattie and Imran Iqbal, who did not seem to mind me asking the same questions over and over again while helping me to sort out various kinds of paperwork. I would also like to send big thanks to all my past and present PhD colleagues at the department, who not only gave me support and feedback but also companionship throughout these years.

This research was funded by LSE's Research Studentship Scheme, KAUTE Foundation and The Foundation for Economic Education for which I am truly grateful. Needless to say this research would not have been possible without these institutions.

I would also like to thank the following persons: Savita Bailur for putting me in contact with the right people to organise my data collection, Joan Komugisha and Nicholas Kamanzi who were kind enough to take the time to give me important insights regarding my research and also to introduce me to other relevant persons, my beloved Emmi Rees for being amazing and for showing me that there are also other areas in the UK besides London, Ali Bushell and Markus Siivola for all the wonderful dinners and evenings, Markus Karhula for lunches, Shona Jane Lee for keeping me company in Kampala, and my family and friends and especially my mom for sending all those packages.

Finally, I would like to dedicate this thesis to Maria Carreras Godall because of so many reasons, not least of them being the thought that it would make her smile.

Table of Contents

List of Figures

List of Tables

1 Introduction

> *"As a conduit for digital information to invade every aspect of modern life, apps are scripts that link the way we search for small software solutions to things that we once took for granted, and to the larger issues facing a hyper-networked society on the precipice of total immersion in the digital culture."* (Matviyenko & Miller, 2014, xi)

The last two decades have witnessed an enormous increase in mobile phone ownership across the globe. This growth has largely come from the Global South, where the decreased cost of handsets has enabled more users to become owners or to otherwise have access to these devices. There are large differences in the types of phones that are being sold, ranging from the so-called basic phones, which allow calling, texting and few additional functionalities, to more sophisticated smartphones that in certain ways resemble more small handheld personal computers (PCs). As prices continue getting lower and in some cases incomes of people higher, there is a move towards these smartphones also among users in the lower income groups in developing countries. In addition, even the basic and feature phones are becoming equipped with more and more functionalities, such as internet access and possibility for installing external applications into them.

This on its part has given rise to new areas of economy, in which business models are built around the mobile phone. One of these areas has been the development of applications for mobile phones and other devices that has led to the creation of globally connected app economies. An app economy has been defined as "*a collection of interlocking innovative ecosystems [, where] each ecosystem consists of a core ecosystem, which creates and maintains a platform and an app marketplace, plus small and large companies that produce apps and/or mobile devices for that platform*" (Mandel, 2012, 2-3). As the definition indicates, an app economy consists of a variety of actors that can participate in the functioning of the app economy in various ways, one key role being naturally reserved for the application developers and technology start-ups which build the applications. In the process of creating the applications, the digital tools needed in the work are nevertheless essential, and those are also the one that make application development different from many other type of technology development. Due to their digital nature, the technological resources needed in application development can be accessed basically from any corner of the world. As a consequence of this, thousands of developers from different areas and locations have taken part in the app economies by building applications of various types and for different devices (Gathigi & Waititu, 2012; Goldsmith, 2014).

Overall, from a technological perspective application development and creation of app economies have become feasible also in the more underdeveloped regions (Gathigi and Waititu,

2012; Goldsmith, 2014; Graham, 2008). Thus, many of the developing countries have jumped into the bandwagon of application development and supported the application development sectors of their economies. Increased reach and efficiency of networks, innovation hubs and inclusive business models are some of the areas that are deemed important in order to promote the IT sectors of these countries, and the general intention has been to raise these sectors to a level where they could substantially contribute to their respective national economies (Graham & Mann, 2013; Meagher, Mann, & Bolt, 2016). Furthermore, countries have published plans on how to nurture their ICT industries and application development environments in a manner that would provide a fertile soil for developing the successful applications (e.g. Uganda Ministry of ICT, 2015; Vision 2030, 2007).

The whole enthusiasm towards application development is captured by a report from Caribou Digital: "*Apps attract significant attention in part because they represent the first truly global market for digital goods, which can in principle be produced anywhere, distributed at almost no cost, and consumed wherever there is a network connection. The low barriers to entry and scalability of digital products thus offer the alluring promise of more accessible economic opportunities, especially to those producers typically marginalised either by socioeconomic status, geographic location, or both*" (Caribou Digital, 2016, 6). The expectation is that application development would not only provide an additional source of income, but also push countries forward technologically and enable the creation of applications that could offer solutions to societal and other type of challenges and problems.

The possibility to create applications and to take part in the global app economy has also led to an interesting paradigm shift in many of the developing countries, especially in sub-Saharan Africa. Traditionally seen as recipients of technology, the digital nature of application development has enabled many of the less developed countries to position themselves as technology producers. In terms of software development, the rise of local application development also differs from the outsourcing activities that are done by multinational and other companies from the developed world: the ownership of the application lies mostly among the technology start-ups and not somewhere else. In addition, the primary market for the locally created applications is at first instance often, although not always, the local market (Wagner & Fernández-Ardèvol, 2016), which means that the primary objective for the applications is to serve the needs and wants that exist in the local markets.

In terms of research, this has opened a new type of research area especially in relation to research on information and communication technologies for development (ICT4D). Whilst the main focus has been often been around issues such as digital divide (Antonelli, 2003; James,

2007), on socio-technical challenges of technology transfer (Heeks, 2002a) or the role of technology in development overall (Avgerou, 2010; Heeks, 2010), the area of local technology production, especially when the target market is a local one, has so far received less attention. One could argue that this is because application development does not differ much from the traditional technology transfer, as the technological resources needed in the work, such as software development kits (SDKs), application programming interfaces (APIs) and also the devices used to access the applications, derive mainly from outside these countries. However, the developers still hold power in deciding what type of applications are built, on which platform and how[1].

In general, it can be claimed that application development in the end is very much a technical exercise, where software developers write a piece of software i.e. an application that runs on particular type of devices, which the targeted users possess and utilize to access the application. Through the usage of these applications societies in general get increasingly technologized as various aspects of people's daily lives are transformed and managed by applications (Cairncross, 2001; Negroponte, 1995). This kind of view is very technology centric, as it focuses mainly on the role of technology and its increasing role in performing almost any type of action. However, at the same time it is widely agreed by scholars that any kind of appropriation of technology, be that in terms of its creation, development, implementation or usage, is highly context driven (Avgerou, 2001; Walsham & Sahay, 2006) as same technologies can have different results and outcomes in different locations and contexts (Adams & Ghose, 2003). For example in software development outsourcing, location plays an important role in determining the results (Abbott & Jones, 2012).

This leads to a question on the particular role of context in application development. Overall, research has been undertaken on entrepreneurship and other related issues in the context of developing countries. There has also been a notable amount of research on the different factors that contribute to the creation of entrepreneurial ecosystems, where studies have tried to find out the legal, institutional and other factors that need to be in place for entrepreneurs and their businesses to flourish (Foster & Azmeh, 2016a, 2016b; Kambhampati, 2002; Molla, 2000). However, the fundamental question on why people build the applications they end up building

[1] Application development in general can be seen as a prime example of generativity (Kallinikos, Aaltonen, & Marton, 2013; Tilson, Lyytinen, & Sørensen, 2010), which can be defined in somewhat different ways (Tilson et al., 2010; Zittrain, 2008), but common to most of them is the notion of a system's ability to harness contributions from external actors, for example in the form of applications, in ways that have not necessarily been foreseen by the system creators. Although generativity as a concept is not among the ones used in this research, it highlights the point made here about using external resources to create applications, and within that process, these resources leave the developers considerable room in deciding what is it that they want to build and how.

and how do those applications get shaped during the process of their development has been left largely unanswered. This carries implications in terms of understanding what types of applications can be expected to arise from particular locations, and how well they might be positioned not only in creating economic benefits but also solving societal challenges.

This research sets to answer these questions, and approaches the topic of application development in developing countries by adopting a more micro-level perspective. In other words, it concentrates on studying the topic from the perspective of the developers and small technology start-ups. The principal intention for the research is to map the lifeline of applications that are built in a developing country context, and to understand how contextual factors affect not only the challenges the applications and the start-ups building them face, but also what gives rise to the applications in the first place[2]. Therefore it aims to go beyond the factors such as available investment for start-ups or functioning institutions that are more frequently discussed, without arguing that these would not be relevant in developing successful applications. Instead, it argues that other types of contextual factors are also important, and so far have gone to some extent overlooked.

The research takes the approach that Donner and others have discussed by using the term *after access* (Donner, 2016; Gitau, Marsden, & Donner, 2010). This notion refers to a moment where access to a particular technology, in their case the mobile, has to a good extent been reached. What remain less understood are the meanings that particular types of technologies carry with them and the shapes they take in their contexts of appropriation. In relation to application development, it can be argued that many of the technological resources are now available for developers across the globe. However, building applications for a particular type of users has also implications for the application development process. As noted above, the tools might be the same, but the outcomes are likely to be different, or as Adams et al. (2003) note, "*each context of technological appropriation is different: the same technology has different effects in different places and situations*" (p.434).

Within this research, the premise is that the difference can be largely traced back to the overall context and location where the application development and application usage takes place. The research argues that application development should be seen as a process of societal routine development, where an existing societal routine, such as shopping or moving from one place to another, is captured by the developers. Within the application development process, some parts or stages of the routine performance are given a technological form, i.e. they are expected to

[2] This is important also for theorizing about more macro-level phenomena, such as how application development for local markets in developing countries might impact these societies overall.

be performed by using the application that the developers create. The difficulties for application development in a developing country context become evident in this process as only a limited number of these routine performance stages can be technologized, while the other relevant stages of that routine still have to be performed manually without resorting to the local technological resources typically available in the more developed regions and countries. In terms of the availability of these technological resources, there are stark differences even between developing countries themselves, where certain countries such as Brazil or China are clearly more advanced in this regard than for example many of the countries in sub-Saharan Africa[3].

In terms of impacts, this is relevant especially to many of the developing countries, as the research enables them to have a more holistic understanding of the deeper contextual factors that play a role in developing these new digital pockets of their economies. Furthermore, although this research takes a more micro-level approach, the findings can be linked back to the macro-level structures of these societies as well. Context overall can be seen as a particular type of structure, within which individual actions are connected to the prevailing and bigger narratives that affect the whole functioning of societies.

Before moving on, there are few clarifications that are worth mentioning. The first one relates to the area within IT sector that this research aims to cover. IT sector or IT industry can be divided into five areas of production that to some extent may also overlap: goods, software, infrastructure, services and content (Desai, Purohit, Stadje, & Wong, 1998; Heeks, 2006; Molla, 2000). This research focuses mainly on software production, and within it, software production that is not outsourced from other countries or regions. Specifically the thesis aims to understand the role and impact of context in relation to relatively small-scale application development, i.e. small technology start-ups. Another point that needs to be noted is that this research studies the topic from the perspective of start-ups that exist in a particular region, namely East Africa, concentrating especially to the region of Kampala, Uganda. As noted, the term developing country captures a wide range of countries which differ between themselves in various ways. In relation to ICT and software development, it is clear that countries like Brazil or India are quite different from countries like Uganda, which has very little if any tradition on software development for example as an outsourcing destination. Therefore many of the results of this research are likely to apply at first instance to countries similar to Uganda but less so to countries like China or Brazil. Finally, the start-ups that have been studied here develop their applications first to the local market. This does not mean that there is no incentive for these companies to scale up their products to other areas at a later stage. However, what is crucial here is that the

[3] This is also linked to the concept of a developing country i.e. how does one define what is a developing country and what is not.

first target market for the applications is the same where the start-ups themselves are based. With the main research question set for this research, "*how does context impact the application development process in a developing country*?", the aim is therefore to understand how are applications developed in a particular location, and how does the location itself affect that process in addition to the technologies used to build the applications.

1.1 Contents of the Research

The chapters in this thesis are organized in a manner that corresponds with the typical process of application development. Application development start with the ideation, where the idea for the application is formed and in some cases developed further for example by seeking some form of guarantees that the idea for the application is valid and worthy of developing. After this, the application in itself is built and other aspects related to the functioning of the application mapped. Finally, the application is launched to the targeted user groups, which either start using the application or reject it. Although this type of description is simplistic as in many cases the different stages overlap and also form a loop where user feedback feeds into the updating of the ideas and the application itself, it is seen that by using this structure, it will also make the thesis and its contents clearer and provides a logical order for discussing the areas that are relevant for the research.

Following this order, chapter 2 discusses the relevant literature and maps them according to their relevance to each of the stages discussed above. It starts off however by looking at the connections that have been found in the literature between information and communication technology and the development of developing countries. From there the chapter moves on to discuss literature on designing products and services in a developing country context, which is followed by studying the literature on technology implementation that occurs in these countries. This topic is relevant for this research as although the applications themselves are created locally, they are based on technologies and technological resources that have their origins elsewhere, often in the Western countries. To cover areas related to the usage of the applications, literature on digital divide is discussed as it provides a basis for understanding the challenges related to application usage in a developing country context.

Chapter 3 sets the conceptual framework and the research questions for the research. Basically application development can be seen as interplay between two different factors, the context where the application is built and (hopefully) used and the technology that is required to build the applications. Therefore the chapter starts off by discussing how context is understood within the confines of this research, and from there moves on to explain what kind of a technological artefact an application is. Finally, the chapter links these two concepts together by using the

theory of Kitchin and Dodge (2011) on the interconnectedness of code and space and their role in the everyday activities of people in different societies.

Chapter 4 presents the epistemological and methodological foundations of the research. This research adopts a case study method and uses a critical realist ontology combined with a mild constructivist epistemology. It sets to unveil the mechanisms that affect application development in a particular location, and aims to do so by using qualitative data collection methods, such as interviews and non-participant observation, and analyses the data by using thematic analysis. Furthermore, the chapter explains what was done to the data and shows how conclusions were drawn from the data analysis.

Chapter 5 discusses the key findings from the data and is organised according to the different stages of application development mentioned above. It also presents the case and the surrounding context where application development occurs. Samples from the collected data such as interviews and field notes are shown to give the reader a better idea of the contextual and technological factors affecting the application development process.

Chapter 6 draws from chapter 5 and analyses the findings from the perspective of the conceptual framework presented in chapter 3. The chapter takes the research to a more abstract level by linking the findings to the key concepts of this research, and by doing so answers the set research questions. It further aims to highlight the key contextual factors that affect not only the application development per se, but also the contextual challenges that stem from the targeted user groups, which have further implications for the possibilities of success of the applications.

Chapter 7 links the data analysis back to the literature review of chapter 2 and discusses the contributions this research aims to make to the existing literature. As the topic overall has received only limited attention so far, the research presents a framework for analysing application development. In addition to this, the results of this research have implications also for the literature on technology implementation occurring in developing countries and to the debates on digital divide. The chapter also concludes this research summarizing the main results and contributions of the research as well as discussing areas for further research and the limitations of this research.

2 Literature Review

> *"Use of information and communication technologies for international development is moving to its next phase. This will require new technologies, new approaches to innovation, new intellectual integration, and, above all, a new view of the world's poor. The phase change from information and communication technologies for international development (ICT4D) 1.0 to ICT4D 2.0 presents opportunities for informatics professionals and offers new markets for ICT vendors. […] Where ICT4D 1.0 - fortified by the 'bottom of the pyramid' concept - characterized the poor largely as passive consumers, ICT4D 2.0 sees them as active producers and innovators."* (Heeks, 2008, 26, 33)

The following literature review aims to critically discuss the relevant literature regarding local software development occurring in developing countries. As a result, it functions as a foundation for the research questions and aims to show why these questions are important and need to be answered both from a theoretical as well as from a practical perspective. The review will show that the connection between technology and context has been discussed from several different perspectives, which all however eventually lead to somewhat similar conclusions and point to the need to better understand how contexts affects application development. This is especially important in relation to many developing countries, which see application development as a promising area to invest in, as in addition to economic benefits application development is also hoped to address some of the societal and other problems these countries face.

The literature review starts off by looking into the ways technologies are expected to be beneficial for the development of developing countries and what kind of impacts can be expected from local software development companies whose applications target primarily local markets. After this, the literature review looks into what has been said of application development occurring in developing countries so far. From there onwards, the literature review follows the stages of application development discussed in the introduction, namely ideation, development of the applications, and finally usage. The ideation stage is discussed in terms of technology design, and how design can be seen as an enabler of development. Technology implementation in developing countries are relevant for this research since although the technological resources needed in the building of the applications leave a lot of room for the developers to create the applications they want, those resources still originate from elsewhere. The final section of the literature review concentrates aspects that are relevant for the usage of the applications, which can be seen largely in terms of the digital divide. In order for any of the benefits gained from technology to materialise, a crucial factor is getting that technology into the hands of the targeted users. The literature on digital divide highlights the challenges technologies and ICTs in particular encounter in making their way to developing countries, and how technology usage overall goes deeper than mere access to certain technologies. Overall,

the key notion that stems from the literature review is how local application development has the potential of representing a shift in research due to the locality of technology production. Application development can be seen as an area, where majority of this production no longer takes place outside the borders of the developing countries but within them. The final part of this chapter weaves together this in more detail and shows the gaps that this research aims to address.

2.1 The Societal Impact of ICTs and Their Relation to Context

Information and communication technologies (ICT) are in general considered as having a beneficial impact for societies across the globe (Heeks, 2010). Investments in ICTs are expected to stimulate economic growth (Britz, Lor, Coetzee, & Bester, 2006; Juma & Agwara, 2006), and although this might happen with some delay (Jorgenson, Ho, & Stiroh, 2005), most research implies that ICTs leads to positive outcomes also in other areas than the economy (Chigona, Beukes, Vally, & Tanner, 2009; Chircu & Mahajan, 2009). These positive outcomes have been linked to the capability of ICTs to allow more efficient connections between geographically distant areas, and ICTs have been said of having had a crucial role in integrating new areas and regions into the global economy (Fong, 2009; Friedman, 2007) by enabling significant reductions in communication and coordination costs resulting from geographic barriers (Forman, 2005). ICTs have in this sense been seen as having the capacity to equalise the opportunities for people that operate in areas that have traditionally resided outside global economic and cultural centres. This is important already from a symbolic perspective, as these technologies have become to signify modernization and modernity, and the processes of connecting areas together are expected to have far-reaching consequences for societies and shaping in various ways their economic, social and other characteristics (Rahim, Pawanteh, & Salman, 2011).

2.1.1 Developmental Benefits of ICTs in Developing Countries

Many of the studies on the benefits of ICTs, especially the ones relating to the economy, have been conducted in developed countries, and the applicability of these results to developing countries has been under some debate. Research has however concluded that ICTs' impact on socio-economic factors is likely to be positive (Harindranath & Sein, 2007), no matter the location, and the eagerness of many of the developing countries to adopt ICT-led development strategies has been seen as largely justified (Duncan-Howell & Lee, 2008; Zainudeen & Ratnadiwakara, 2011). The interest towards studying the actual impact of ICTs in for example meeting developmental goals has risen recently, and whereas scholars initially paid more attention to issues such as ICT readiness, availability and uptake within particular regions, the focus has shifted to also studying the impacts ICTs may have on economic and other socio-economic factors in developing countries (Heeks, 2010; Heeks & Molla, 2009). These impact

studies have linked ICTs to beneficial contributions for example in areas like entrepreneurship (Donner & Escobari, 2010; Jensen, 2007), women's empowerment (Khan & Ghadially, 2010), and in general increasing the choices available to people in developing countries as advocated by the capability approach (Kleine 2010).

This does not however mean that success would follow automatically from ICT implementation as it remains clear that many of the ICT projects in developing countries fall short of their objectives and end up failing (Heeks, 2002a, 2010). ICTs' capability to create further development to different regions in the world is dependent on a number of issues, and overall the relationship between ICTs and development is complex and also varies from one place to another. Non-technological factors such as institutional settings, educational levels and general infrastructure are equally important in making sure that the full benefits of ICT can be reached (James, 2007; Kimaro & Nhampossa, 2005). Furthermore, it is not always the case that same technologies would automatically lead to same impacts across regions. Technologies are appropriated differently depending on the implementation location, and those locations pose distinct requirements for the technology to be properly functional. As a result, not all technologies are therefore equally suitable for every context or location, nor can be expected to have similar societal and other outcomes.

One example of this can be found from mobile telephony. In many developing countries mobile technology has been found to have the most capacity to enhance economic growth. This is especially true in countries where prior to the widespread use of mobile phones landline telephony was not largely available, and the difference in connectivity between and after the spread of mobile phones was very large. As a result, in places like sub-Saharan Africa mobile telephony has been shown to have the greatest effect in low-income countries (Swanson et al., 2006; Waverman, Meschi, & Fuss, 2005). On the other hand, Internet has not been able to cultivate similar economic growth in these countries. Its impact in the African continent has been most clear in upper-middle-income countries, but in relation to the low-income countries in the region the economic impact regarding Internet has been close to nothing (Chavula, 2013). The reason for this has been in the linkage between devices and their usage patterns. Although many of the mobile phones afford Internet access and the ownership rate of those devices in developing countries keeps getting higher, accessing the Internet is not among the main usage patterns of mobiles. Furthermore, other ICT devices, such as PCs and tablets, which do enable Internet access, are not that common in low-income societies (Donner, 2016). What follows is that despite the existence of devices enabling connections to the Internet, usage remains low, which also means that the opportunities provided by Internet become less attractive for the

relevant actors and discourages them from actively building web-based services for the local markets.

This example also goes on to show that the term ICT captures various technologies, ranging from mobile phones to internet and software applications, which have all been put under the same umbrella. In addition, many of these technologies are being further developed and keep on changing their form, in a similar manner to completely new technologies that are being brought into the markets and also fall under the term ICT (Heeks, 2010). One example of these are smartphones, which have only existed relatively short time but also keep on constantly changing in their capability to perform different types of functions (Donner & Escobari, 2010). As a result, when discussing the benefits of ICT it is often forgotten that technologies themselves are dynamic as they keep on changing and incorporating new functionalities.

However, it is the contextual factors that appear to affect mostly technologies' capacity to deliver the desired outcomes, almost irrespective of the technologies in question. Same technologies may thus have very different impacts and implications depending on the context where it is appropriated. Therefore, to understand the full impact of ICT in a given location research has to adopt a broader view of technology, where in addition to the technological characteristics, the different social and other contextual factors need to be included into the analysis as well. This is the case not only regarding research on ICT and its impact on economy, but also in other areas where ICTs are expected to contribute positively (Byrne, Nicholson, and Salem 2011; Friederici, Ojanperä, and Graham 2017). Common to many of the studies on the benefits of ICT, be they on health information (Kimaro & Nhampossa, 2005) and geographic information systems (Puri & Sahay, 2003), food security programs (Masiero, 2015) and social activism (Schejter & Tirosh, 2012), is their socio-technical approach, which highlights the role of contextual factors in technology appropriation. The local settings and conditions are an integral part of the functioning of any system or technology, and for a system or technology to have an impact, the context within which it is going to be used must be understood first. This is even further highlighted in research where technology in itself is seen more as a tool but not a deciding factor of outcomes as such. Toyama (2011) for example argues that technologies should be seen as amplifiers of impacts, which can be either negative or positive. The direction an impact shall have is dependent on the underlying context where the technology appropriation takes place but not the technology itself.

Overall, research on ICT and its developmental benefits has also shifted during the past decades, which to some extent can be seen as a consequence of Sen's (1999) capability theory gaining wider attention among development scholars. Instead of focusing on rather static

measurements like economic growth, capability theory takes as an objective for development the increasing of freedoms people value and provides a normative framework for discussing what is meant by human development (Robeyns, 2005). Rather than only looking for example how well a particular country does according to certain developmental indicators, the focus should be in the individuals and groups capabilities and opportunities to live the lives they see value in (e.g. Ibrahim-Dasuki, Abbott, & Kashefi, 2012). Consequently, the notion of development becomes relational as not all the people share the same aspirations, and one's position to reach those aspirations depends on person's capacities but also on the societal position that person occupies (Smith & Seward, 2009). The main point however is that capability approach offers a rather complex and multidimensional version of what is being understood with development, and in this sense contrasts with certain other developmental theories that view development more as a sum of few, often economic, indicators (Smith, Spence, & Rashid, 2011).

As a result, the implications of ICTs have become more varied when it comes to measuring their impact on development, and different scholars have studied the role of ICTs in relation to people's capabilities as guided by the capability approach (Johnstone, 2007; Kleine, 2010; Zheng, 2009), and Sen himself has for example discussed the role of mobile phones in enhancing people's capabilities (Sen, 2010). Overall, mobile phones' main implication for development has been seen in enabling access to relevant information and increasing connectedness between people (Smith et al., 2011). This type of development is not only linked to economic development but to other areas as well. As a result, the question on what is measured becomes more varied, as other factors alongside the economic ones are also taken into account. Furthermore, research on technology should not cover technology alone, but in addition areas that are closely linked to it. One example of these areas are innovation and innovation centres such as technology hubs that exist in developing countries, which in addition to technology development can be seen as having a role in boosting entrepreneurship and also in providing other than economic capabilities for the people that reside in them (Jiménez & Zheng, 2017).

Technology therefore has a multidimensional role in relation to the development of different regions and countries. It can be used to deliver outcomes of economic nature but also, as the literature has shown, of improvements in various other areas and aspects. As noted by capability approach, the question on what is considered development is a relational question, and in that sense largely contextual. However, it is not only that ICTs may bring with them changes that can be seen as positive from a particular perspective or in relation to a certain variable. The question that has been more difficult to answer is in which ways context affects ICT deployment and

usage, and also in which manner, if any, technologies may themselves have an impact on the locations and contexts where they get implemented (Heeks and Molla 2009).

2.1.2 The Effects of Technology on Context

Most scholars do recognise the importance of context within ICT appropriation or practically in any area linked to ICT, be that design, development or usage. However, another question that has been asked is how technologies themselves affect the locations where they are implemented and used. Walsham notes how there was a tendency to view ICT-enabled globalization as something that over time would lead to the homogenization of cultures and societies (Walsham, 2002), which would have rendered different contextual factors to having lesser importance. Similarly, some argued that increased ICT usage on a global scale would cause the death of distance (Cairncross, 2001), and as the geographic barriers or the limitations of being tied to a location would disappear, different geographical areas and societies would resemble each other to such a degree that also the importance of context in technology or ICT usage would lose its meaning[4]. As a consequence, same technologies could be used in different locations with similar results, no matter the context or place (Greig, 2002).

These views have been largely contested by other scholars (Graham, 2013; Wang, Lai, & Sui, 2003). Morgan (2004) for example argues that spatial reach is not the same as social reach, and merely concentrating on the spatial forgets the importance of the social: being able to pass information from one location to another does not mean that understanding follows as well. Also despite the advancements in ICT, physical proximity still enables more effective communication, something that can be seen in companies placing importance to physical proximity in the organizational design of complex activities. Furthermore, location and factors related to it remain important for companies regarding their outsourcing decisions (Abbott, 2013). Similarly in relation to innovation and other activities, if location truly did not matter, these activities would have spread more evenly across the globe. The bottom line seems to be that simply put people use and understand technologies differently. Robertson (1992) describes this by noting how technologies go through a process of indigenization in their locations of appropriation. Some technologies and the ideas they carry with them get rejected entirely, while others, in case they are accepted, are then adapted so that they better fit the local context. Overall, ICT facilitates the collision between communal, societal and global values and customs, yet the end results vary from one place to another. Different histories, geographies and

[4] Ritzer (2007) also discusses this homogenization in relation to his concept *'McDonaldization'*, in which he sees that by entering new markets multinational corporations bring with them certain thought models related to efficiency, calculability, predictability and control, which then affect the local customs and work habits of the receiving societies in a largely similar manner, little matter the location.

languages are likely to lead into differences in technology appropriation, and the diversity in values, customs and contexts leads to distinct understandings and usages of the seemingly similar technological artefacts and tools (Appadurai, 1996; Walsham, 2001).

However, whilst it has not been the case that technology has led to the disappearance or diminishing importance of context and contextual factors, it cannot be argued either that technologies overall would have no impact on societies. ICTs are been seen as transformative in their capacity to alter two central actors in human life, time and space, as these technologies reshape our understanding of these two concepts. These technologies enable people connect and contact each other instantly, access information at any time and as a result, restructure for example the ways people work, interact and spend their free time, all of which also affect the societies at large as well. Overall, societies that become increasingly connected and also dependent on ICTs are likely to go through different economic, societal and cultural changes (Castells, 2011; Lee & Whitley, 2002). As pointed above, this does not necessarily mean that the changes will be similar everywhere, and for example knowledge of the global influences that follows from being connected has been claimed to make local communities and societies more aware of their own characteristics and to take effort in preserving those (Hongladarom, 2002; Robertson, 1992). The result appears to be that societies do change because of ICTs, yet in a manner that enables them to keep their 'identity' while being connected to other areas and regions. In a similar vein, ICTs do not make automatically the current notions of time and space irrelevant as those still function as the principal way people in societies organise their lives (Green, 2002).

However, technologies when being actively used do pose a particular way of functioning, and as different customs and habits get incorporated into the usage patterns of technologies, they also shape the way societies and the people within them function in different situations (Latour, 1990). This is similar to Lessig's (2006) argument on the regulatory notion of code, in which he sees different software as guiding the behaviour of the people who are being subjected to the usage of that software. The software does this by allowing only particular ways of functioning at the expense of others, which can be seen for example in the different standards that are given a technological form in software programmes. Through the usage of those programmes the standards become inscribed into the everyday lives of the people, forming the new normal. Therefore, the way ICTs are being deployed and how they form a part of the functioning of societies at large has implications for what constitutes as the standard mode of operation or practice in situations, where technology plays a part in carrying out that practice or operation.

This process is further strengthened by the innate nature of technology and how it is developed. Technologies are often created to perform functions and tasks that exist within societies or organisations, and in this process these tasks and functions get simplified as the technology is unlikely to capture all the richness that exists around these tasks and functions when performed without technologies (Kallinikos, Hasselbladh, & Marton, 2013). This in its part further leads to having less flexibility in performing the tasks, restricting and limiting the options available for the user. Just as technology can be beneficial in creating social and economic benefits both for developed and developing countries, it also tries to pose its own particular logics and rationales to the situations and societies where it is being used, and in some cases also functions as a tool to promote standards that have been seemed rational in the countries of origin of these technologies (Ciborra, 2005).

From a theoretical perspective, there is an interesting tension between the notions on technology and context. As described above, technology has been considered for the most part of being a force of good for the development of developing countries, although its appropriation and impact has been seen as being conditioned by various contextual factors as well as by what is understood with context. However, simultaneously it has been seen as coming in with its own set of standards and operating models, which are likely to change these contextual factors if successfully implemented. As a result, they appear to form a tightly intertwined relation, where both affect the other and in ways that are not entirely known. Traditionally this however has not been much of an issue in developing countries, where technologies in many societal areas have remained scarce. More often the problem has been seen as being the opposite, as underdevelopment has sometimes been linked to lack of technology, which was demonstrated for example by the one laptop per child (OLPC) initiative (Irani, Vertesi, Dourish, Philip, & Grinter, 2010; James, 2010). Despite this, there have been some discussions on how developing countries themselves could become more active in producing technologies of their own. As noted above, application development has been seen as an especially promising area for developing countries to contribute in.

2.2 Software Development for Local Markets

Easier access to technological resources has been seen as a tool to democratize content production in different parts of the world, and as it is done locally, the expectation has been that the content would correspond better with the interests of the local users (Powell, 2001). Overall by designing and producing items and artefacts locally, they are at least in principle more likely to be aligned with local realities, and as a result, have more success among the targeted users (Heeks and Molla 2009). This is crucial as in order for an area to reap benefits from ICTs, not only

local production of ICT-based good and services is necessary but also their consumption (D'Costa, 2006).

However, there are indications that despite the more equal access to technological tools and resources, other structural obstacles impact countries' possibilities to take part in the global markets of application development. As an example, the mobile application markets are heavily biased towards Western countries and the whole business is largely controlled by the major US-based companies. The resources needed in application development such as SDKs and APIs often have their origin in the Western countries (Caribou Digital, 2016), and it is not entirely clear whether these tools and resources enable local content creation in a manner that matches more accurately the needs and wants of local communities. In addition, the willingness of the developers to create content for local markets is not automatic but depends largely on the incentives. Wagner and Fernández-Ardèvol (2016) found in their research on app developers in Bolivia and Argentina that many of the developers were not too eager to produce content for the local markets as the home market was not seen as a very lucrative one, although the situation was slightly better in Bolivia, where aid organizations were promoting app creation to meet local needs and wants.

Furthermore, local mobile application production does not automatically substitute global offer either, although does reduce it to some extent. In general the market for applications is distorted geographically, and there are differences in application production across countries and especially in relation to revenue. Outside the few top regions, very little revenue goes to places like East Africa, which has been said as being result of lack of demand for local apps, or in some cases, barriers like language, which make it easier for outside applications to enter the local markets. Also platform design has implications in terms of deciding which countries can reap the benefits of application development. Some of the mobile application stores do not allow developers from certain countries to monetize on their apps, and for example both the app stores of Apple or Google have geographical restrictions in place regarding monetization (Caribou Digital, 2016).

In sum, the relatively easy access to the tools needed in application development has not made the playing field entirely equal between different countries. The arguments for local production state that the general advantage in meeting the local needs and wants should reside among the local developers, as they are more likely to know the context. This might be true, but in terms of the finished applications, it appears something gets lost during the development process.

2.2.1 The Variables Linked to Software Industry

One reason hindering the success of the local applications could be found from the technologies themselves. Many of the technologies used in application development in developing countries derive from somewhere else, which creates a type of dependency between the creators and the receivers of a technology. From the perspective of the receiving countries, it is seen as important to contradict this type of dependency by building their own ICT infrastructures and produce their own ICT enabled services and products. Otherwise they become victims of what has been called as ICT enabled imperialism, where certain countries and regions become dependent on the knowledge of others (Ya'u, 2005). Traditionally much of innovation and research have concentrated to richer countries and left many of the middle and low income countries aside. As a result, these innovations tended to address issues that were relevant mainly to the rich countries and much less so for the poorer ones (James, 2007; Singer, 1970).

Compared to many other areas of technology production, software development in general offers more freedom for the developers to decide what is it that they will build and how. In addition to being able to programme the applications of their own liking, the developers can also choose the platforms on top of which they build their applications. Therefore they can be claimed to have a relative high level of autonomy in relation to the platform owners when choosing the platforms. It is also noteworthy that the foreign companies behind these platforms and resources can also be helpful in providing technological and other kinds of support. In practice this can mean funding, technological resources, or just helping the local actors to have the necessary tools and skills to scale up to new markets and users (Kshetri, 2016).

Overall several variables must be met a region or a country to make the most out of the easily accessible technological resources. Connectivity is one condition but not sufficient in itself (Ojanperä, Graham, Straumann, Sabbata, & Zook, 2017), and also mere access to technological tools alone is not enough for a country to create a thriving software sector. Other factors such as skills, competition between companies, clustering and government policies have also been seen as playing a role in the process (Heeks, 2006). In sum, infrastructure, level of education, investment, research and supportive institutional environment are all examples of the type of areas that need to be taken care of in order to create thriving environments for local application development (Kambhampati, 2002; Molla, 2000).

These factors that need to be met for technology production and innovation systems to flourish have received some attention in the literature. Studies of this type have often a quantitative basis, where countries' readiness and possibilities for the creation of a software development or some other sector of economy are evaluated on the basis of certain key variables (InfoDev,

2014; Karippacheril, Nikayin, De Reuver, & Bouwman, 2013; Murenzi & Hughes, 2006). Each region has also characteristics such as local market size, which might be more difficult to change and at the same time set different regions apart from others. In addition, national policies, even when they are not directly linked, may affect the aspirations of a region to participate in the digital economy, and also varies from one region to another (Foster & Azmeh, 2016b, 2016a). Differences in these type of factors may lead regions to adopt and perform different tasks within the digital economy, varying from business process outsourcing to content creation (Mann & Graham, 2016). On a more individual level, one's socioeconomic characteristics, such as education, race or gender, have been linked with person's willingness to take part in content creation, and in general those from more privileged backgrounds are more likely to produce online content than people coming from poorer and more marginalised areas (Schradie, 2011).

In general, much of the literature on local application development has concentrated on the mainly macro-level systemic characteristics and structural factors that are expect to have a positive impact on the functioning of the local innovation systems and with it, application development as well (Rai, Harindranath, & Liebenau, 2013). Context in these studies displays itself as a combination of the variables and characteristics that are being measured, such as level of education, infrastructure or governmental policies. This type of research is of value when trying to imitate the success of some countries and also to avoid the failures of others. Overall, many of these factors appear to be rather similar to the ones made in literature regarding outsourcing, which has mapped key location-specific factors that affect companies' decisions to outsource some of their functions to other locations (Abbott, 2013). Regarding application development for local markets, there is less knowledge on understanding the deeper effects of context and answering questions on what types of applications are built and why, or how is it to build applications in a developing country context. Without having a better view on these issues, it is also difficult to understand fully the benefits local application development may have for example in terms of matching their applications with local needs and wants.

In sum, there has been limited amount of research on the topic that has taken the perspective of the developers, but calls have been made in the literature for that to occur. Xiao et al. (2013) point out that not too much is known on ICT innovation occurring in emerging economies. They identify several gaps in the literature that require more research, one of them being the need to investigate the design or creation of innovations from the perspective of individuals that are involved in the process. Furthermore, they argue for the incorporation of the local context into relevant theories, and instead of using slightly modified theories that derive from western thinking, researchers should aspire to create new theories that better fit developing country contexts. In addition, research should also try to cover technological areas that so far have

received less attention, such as mobile-enabled or cloud-related innovations. Regarding research objects, Morgan (2004) states that research on innovation in general should go beyond the institutions and pay more attention to the role of firms, since those are at the very core of the innovation process. Better understanding of context is also an area where ICT4D research can contribute into information systems research, which has already occurred for example in terms of implementation and usage studies (Avgerou, 2008). More recent research from practitioners has made similar calls regarding application development, i.e. to have more in-depth studies to further understand the role of context in application development (Caribou Digital, 2016). The issue is also more current due to the spread of internet, as more and more people are starting to adopt more than just voice services in developing countries (Zainudeen & Ratnadiwakara, 2011). This further resonates with Walsham and Sahay (2006), who stress the importance to analyse how people in the developing countries use the resources available for them through the Internet, or if they are able to use them in the first place due to lack of skills and access.

This research aims to answer these calls by looking at application development for local markets that occurs in a developing country. Application development can be roughly divided into three separate stages, where the first one consists of the ideas and designs for the application, the second one building the application and the final stage is about the usage of the application by the targeted users. Following this mode of thinking, this research has aimed to identify literature that is relevant for each of these stages in relation to the topic of the research, application development in developing countries. As a result, the first part of the literature review below will look at literature on design that is relevant within a developing country context. Regarding the second stage, the actual development of the application, literature on ICT implementation studies is discussed; although the technological resources allow considerable room for the developers, these resources mostly originate from elsewhere and need to be appropriated or implemented into the work patterns of the developers. Finally, a key area in literature that deals with usage of ICT is related to digital divide, which is also something that the local developers need to take into account when building and launching the applications.

2.3 Design as a Contextual Process

Many of the problems that technologies face in their implementation and usage can be traced back to their design. Different locations pose different challenges, and as an example Akrich (1992) describes how the design of one particular technological object, the lightbulb, was adequate and seen good in Europe, but when exported to Africa, was considered as inadequate and unable to meet the needs and wants of the locals. The European designers had come up with a design that hid the technology from the users. Although this was very much to the liking

of the European consumers, who only had to plug the bulb in and see it working, for the Africans this same characteristic meant that the object was impossible to hack and adapt it to the local conditions in which power sources varied, making it practically useless in the African context.

Design of artefacts does not exist in a vacuum but is affected by various factors. Ideas and existing factors can be designed into artefacts, and there have been claims that also on a larger scale, concepts like socio-economic development can be viewed as design processes (Little, 2004). Overall, design often captures within it historical, cultural and political trajectories in a manner that easily goes unnoticed. However, by acknowledging and becoming aware of the links between context and design, design processes can also be started with the intention to incorporate particular modes of thinking into them and make designs that are more useful in reaching particular developmental as well as other targets (Hoven, 2012).

2.3.1 Development as a Design

Oosterlaken (2009, 2012) states how theories on development such as capability theory can be given a more tangible form by including them into general design principles and through those into actual products and artefacts. Just as design takes the contextual factors that are present in the design location and get transferred into the object by the designer, it can also be made consciously to capture notions of change by incorporating values that represent that change into the design (Latour, 1992; Winner, 1980). This links back to the point of technologies as carriers of values and particular ideologies (Winner, 1980), and for example in terms of capability approach, technologies and their design process can be shaped in a manner that they become tools to expand capabilities (Robeyns, 2005).

Overall, design with an objective to enhance development, however defined, is discussed in literature under various headings. One of these, design for development, was used originally to describe the creation of relatively low technology products that were often prevalent among the poor (Papanek, 1983, 1986). Recently the concept of design for development has been used in national development policies, which argue for the need of developing countries to adopt design capabilities. The rationale behind this is that instead of just being manufacturers of goods and products designed elsewhere, developing countries could enhance their influence and gain a better position within the global economy by adopting and becoming centres for design themselves (Bonsiepe, 1991; Margolin, 2007). This type of thinking marks a shift from poverty reduction to a more economic approach, which stresses the need for these countries to create products for exports and solving some of their problems through using the mechanisms of the market economy.

Similar to concept of design for development, appropriate design has as its starting point the notion that different social powers play a significant role in design. Traditionally these social powers have favoured entities and countries that have held relatively powerful positions in the global markets such as industrialised countries (Nieusma, 2004). Appropriate design shares common features with design areas such as feminist design (Rothschild, 1999), ecological design (Ryn & Cowan, 1996) and participatory design (Cherkasky, 2004; Schuler & Namioka, 1993) as all of them stress the point on how design can become a tool for change. Appropriate design therefore argues for diversity in design and to some extent appraises disagreement as a tool for designers to become aware of the political and other dimensions their works hold within (Nieusma, 2004).

For a design to be beneficial for development and developing countries overall, a crucial factor has been in understanding how contextual factors affect the usage of the designed products and services. For example in relation to designing for the poor, scholars have argued that since the design process and design as a field of study has been largely driven by markets, designers have not given much attention to the methods and structures behind design and design work, and as a result, methods that could promote a more social design have gone largely unnoticed from the designers (Margolin & Margolin, 2002; Thomas, 2006). However, more recently others have noted that this has started to change (Oosterlaken, 2009), and designers and design researchers have become more aware of the other dimensions design as a research field may have. One clear indication of this have been the calls to include the actual users of the applications and products into the design process, and by doing so making sure that the design also tackles issues that are seen important by the users and not just those envisaged by the designers (Dearden & Haider Rizvi, 2015).

The question has therefore become one on the factors that have to be taken into account for a design to be beneficial from a developmental perspective. Many design initiatives that have their basis in engineering have often taken a technological rationality approach and put aside most of the non-technical factors. According to Nieusma and Riley (2010), this has been problematic as engineering initiatives with a developmental goal have tended to fail due to three reasons. First, they have not considered the non-technical factors which play a role in appropriating technology and as a result neglected the social power relations that affect the interaction for example between development workers and members of the community, which may have implications for the initiative's perceptions and intentions of usage. Second, the initiatives are often based on policies such as neo-liberalism, the rationales of which may not fit the local context and make it less likely for the initiative to work. Finally, there is the question on sustainability, since by not considering the non-technical factors the initiatives become easily

unsustainable on the long term as they may lack for example the backing of the local community (Nieusma & Riley, 2010). As a result, any kind of design must also understand the non-technical factors affecting usage, and if the aim is to for example reduce poverty, the design must be economically, environmentally and institutionally sustainable and must have a social goal incorporated into it (Thomas, 2006).

Especially in technology driven projects, the main objective has in the past become easily one of providing large amounts of technology to a particular location or institution, but less attention has been given on issues like how the technology is actually used or its ability to meet the overarching goals that the projects hope to achieve (Zheng, 2009). In this sense placing a design within a theoretical framework or concept, such as capability approach in development studies, makes it easier to shift focus from a more technical measurements, such as the amount of technology transferred, to actual impacts that a technology has in the location of its implementation (Madon, 2004). Overall, technologies or ICTs in general should be viewed as one part of the solutions different design outcomes try to provide. The design of a particular service, policy or an artefact linked to development is unlikely to succeed alone, but design should be done in conjunction with other actions that enable the design to meet the expectations placed upon it (Walsham, 2017).

This is especially important as design is often understood in terms of artefacts and services, but as mentioned above, also the more macro-level events and policies can be seen as outcomes of design processes. Little (2004) for example notes that design as a concept is relevant in understanding the ways different regions in the periphery connect to the global economy, and for him the process of integrating one's economy to the global one is a matter of design, where certain regions have historically done better than others. He sees that design should be seen as broader than only in the terms of designing an artefact or any kind of item, and instead design is something that occurs also in areas like policy development. Thus, reasons why certain societies end up the way they do can be seen as a matter of design. For example, policies are often based on contingencies that are connected to the prevailing political and cultural characteristics that are in place in a particular region, which further contribute to differences in design decisions and development outcomes.

Therefore, any kind of design decision is affected by the environment within which it is born. As has been pointed out by Bijker and Pinch (Bijker, 1995; Pinch & Bijker, 1984), the design process of an artefact can be traced back to social values and events that have an effect on what the designers and decision-makers see as desirable and appropriate. Technologies themselves can be seen as results of the locations or entities where they are created, as the different actors and

stakeholders like designers, engineers and users, who are relevant in the design process, transform their ideologies and beliefs into the products they create, although in some cases this occurs to some extent subconsciously. To fully understand where certain technologies or the ideas behind them come from it is of importance to study the contextual factors and historical settings that lead a particular technology or artefact to be designed, developed and used in the manner it is done (Ananny & Winters, 2007). Design outcomes and the possible success of innovations can be seen as results of context dependent trajectories, where even many of the newly adopted technologies have their origins in the traditions or customs that pre-exist the technologies and continue to shape their usage. A new design or innovation rarely stems out of nowhere, but moreover forms an additional step to a line of other events and processes that precede it.

2.3.2 Design as a Result of Path Dependencies

These trajectories can be studied from the perspective of path dependencies. In relation to technological innovations, path dependencies can be seen clearest in the form of incremental innovations, where new innovations and designs are built on the foundations or on top of existing technologies. Technological changes can therefore be claimed to be largely pre-structured, where the actual change in a technology stems and forms a continuum with the previous technology as well as with the social processes that affect the creation of innovations (David, 1985; Dosi, 1982; Foray, 1997). Furthermore, institutional forces and factors such as the economic notion on increasing returns can make these type of path dependencies even more stable by automatically discarding options that would otherwise be considered plausible and worth taking, thus leading sometimes to the rejection of adopting completely new types of technologies (David, 2001). Path dependencies have been seen as causing a particular type of determinism, where an update or product change almost automatically follows another in a relatively known form. The question has been then, taken that the creation of new innovations and designs is not deterministic, what causes a new technological path to emerge so that the designers deviate from the current trajectory to direct the technological development to new areas.

One explanation for deviations from existing paths has been found from the multi-dimensionality of the concept of path dependency. Path dependencies are not only technological but are also affected by the non-technological factors (Schienstock, 2011). Especially within organizations the designers are constrained by the material, cognitive and organizational layers that are in place in the surrounding context[5]. Henfridsson et al. (2009)

[5] The material layer refers to the tangible outcome of a design, which is the part that can be touched, seen or used, whereas the cognitive layer functions as the mental schema for a design or as a sort of

argue that all these layers together form the dominant design logic of an organization and create dependencies in the design process. By being separate the layers leave the designers room to challenge the constraining factors that stem from the layers, and create new design paths with new configuration spaces thus making it possible for new type of designs to occur. The possibility to challenge the existing design path has its basis in the contradictions that exist between the layers, as all three are dynamic and keep constantly changing but not necessarily in a parallel manner that they would point to the same direction.

However, others have noted that organizations may just as well become locked to the existing paths and are satisfied with mainly incrementally improving the existing technologies and products. Developments and advances for these products are therefore made without actively looking for alternative technologies that could be used, and as a result, organizations become trapped to the technologies they rely on (David, 1985; Dosi, 1982; Nelson, 1982). In addition to technology, also factors like strategies, business models or services that are linked to the technology may create similar type of path dependencies (Markides, 1997). These are often referred to as innovative path dependencies, where the whole innovation processes are seen as following a certain path (Thrane, Blaabjerg, & Møller, 2010). Path dependencies therefore have their basis on both the technological developments and trajectories, but also in organizational factors, where areas such as knowledge and skills play an important part in deciding the set of possible directions that an organisation can take.

Hanseth and Lyytinen (2010) also note that path dependencies work differently depending on what is being designed and therefore have different implications for example in the designing of IT functionalities, applications, platforms or information infrastructures due to their different levels of complexity. Among other things, with increasing complexity the number of actors involved in the process also tends to go up, which makes the variations from the accustomed paths more likely. In addition factors that are often linked to more complex designs, such as incompleteness and open-endedness of large infrastructures, have implications for how these systems should be designed. Applications and to some extent platforms form a sort of middle ground in relation to path dependencies as their area of usage and overall purpose is easier to control. Applications for example have been seen as being the result of transforming the identified user needs into a technological form according to the capabilities given by the particular technological resources that are being used (Freeman, 2007; Walls, Widmeyer, & El Sawy, 1992). However, as systems become more complex, like in the case of platforms, it

reasoning that is put in place for the different functionalities of a design. The organizational layer on its part establishes the processes according to which the design work gets performed, and basically dictates the roles and ways of working within the organization.

becomes more difficult to define in detail the exact objectives for a particular design, as with complexity those may easily change during the design process (D. S. Evans, Hagiu, & Schmalensee, 2008). Therefore path dependencies, at least in the design of technology, should not be seen as deterministic but more from a probabilistic perspective. Any design of technological artefacts and systems is an outcome of constraints and possibilities stemming from the social and organizational environment where the innovation occurs as well as those put in place by technological factors (Hanseth & Lyytinen, 2010). In sum, the design of technological artefacts can be linked back to the properties of technologies, but also to the environment and context they are being designed in, as the social and economic factors affect the usage and purpose of the created artefacts and systems (Walls, Widmeyer, and El Sawy 2004). The determinism of path dependencies diminishes with the increasing levels of complexity of the designed artefacts, but despite complexity this determinism is to some extent always present and favours certain design choices over others.

Traditionally in relation to many developing countries, technologies have been imported from outside these countries, and as a result the technology has not had any particular societal paths preceding it. Therefore it could be argued that there is a discontinuation in the technological and social trajectories. Overall technologies are expected to take certain design paths, but if there is little that precedes them, the question arises on how a path forms or what path is followed. Braa et al. (2007) have argued for a need to introduce flexibility to the standards that technologies are built upon. This is something that happens in relation to application development, where the technological resources allow considerable room for the developers to build the applications. However, importing technologies that leave space for manoeuvrings in terms of redesign can also cause radically different design outcomes, as many of the contextual factors that affect design choices and decisions become replaced with the ones that prevail in the new location and context.

Path dependencies also exist on a more macro-level, as different actors, cultural and institutional settings, technical systems and knowledge bases, may reinforce each other and suggest particular design routes, even for technologies originating from outside. Therefore these external technologies may not have a specific path to take in their new location, but other path dependencies that are in place in the location also affect their appropriation. At the same time, a new technological paradigm may alter the existing paths and with it the design and innovation procedures within that location and open room for alternative paths. These new paths are then results of either changes in the technological paradigm or alternatively triggered by relatively small events in other areas of the design paths, or a combination of both (Fuchs & Shapira, 2005; Schienstock, 2011).

Overall, what this implies is that there are several different paths existing at the same time, and that those may collide in a manner that causes changes to the technological and design trajectories. For this reason Djelic and Quack (2007) prefer the notion of path generation over path dependency, as the former leaves more room for new types of paths to occur. How these existing paths and contextual factors impact design has not been fully answered. Local application development occurring in developing countries provides a case to study this, as many of those countries have little if any tradition in application or software development. Therefore the design of these applications can stem from various sources, starting from the technological constraints and possibilities offered by the tools that are used to build the applications and the design processes adapted from elsewhere all the way to the local contextual factors.

Avgerou (2010) notes that in terms of development, ICT applications can have one of two types of impact. Progressive applications deliver substantial impacts by creating new income or capabilities, yet keep the prevailing deeper structures and mechanisms intact, and in this sense follow the existing path dependencies. Transforming applications on the other hand can be seen as altering these as they have more fundamental impacts in terms of creating new business models and even affecting the power positions that exists in a particular society. Determining to which category local application development and design is more likely to fall is not straightforward, as it can be seen as being located in the crossroad of the two sometimes contrasting forces, local contextual factors on the one side and the technological tools and resources on the other.

2.3.3 Application Development as a Socio-Technical Process

It has been noted that in terms of technology adoption in a particular location, historical factors such as legacies of other, already existing technologies affect the adoption of new technologies (Zhu, Kraemer, Gurbaxani, & Xu, 2006). In relation to application development, in most developing countries there is not much tradition for it, and neither is it entirely clear which other technologies might impact the adoption of SDKs and other technological tools and resources needed in the process. In relation to the path dependencies and technological trajectories, this has the potential of enabling the designers to come up with completely novel ideas and designs. Despite the constraints that the technological resources place, they also leave some room for the developers to develop the applications they want and to combine these resources in countless different ways.

The same applies to organizational path dependencies. The start-ups building the applications tend to be relatively new and have likely limited amount of established organizational patterns,

which would create design related dependencies within the organization. Therefore the design is likely to be more impacted by the external conditions that can be found outside the start-ups. New design paths are to some extent always shaped by and built on the foundations of what exists in a particular location (Kemp, 2002). The location therefore further stresses the point that despite the ubiquitous nature of the technological tools used in software development, context and with it, location, affect the creation of software.

In general, innovations are seen as getting their meaning when they become part of social practices of their users (Tuomi, 2002). Similar notions have been made on application development, where developers form a feedback-loop with the users and use the inputs gotten from the users to further develop their applications (Fontana & Sorensen, 2005). Also in cases where there is no direct interaction with users, the developers receive feedback from their users in the form of reviews if the applications are distributed via app stores that enable user reviews (Iacob, Harrison, & Faily, 2013). User feedback, be that in the form of direct interaction or reviews, also means that even if the developers themselves were detached from the location, the users would be able to impact through their opinions the incorporation of contextual factors and their implications for the design. As a result, also technologies that can be seen as somehow new are likely to have inputs from the surrounding environment where they are to be used, and therefore also have connections to the local contextual factors. Overall, with the increasing amount of software that is being used in different parts of the societies, it has become clear that in the end, software development is always tied to the requirements set to it by the societal and other social factors (Fuggetta & Di Nitto, 2014).

This can already be seen in the technological choices that are taken regarding the applications. For example for mobile applications, one of the crucial decisions is to decide on which platform or platforms the application is built on. The more platforms the developers choose, the more complex the software development becomes, already in terms of design and application updates which then need to be done for each platform separately (Joorabchi, Mesbah, & Kruchten, 2013), although some possibilities for developing to several platforms by using same tools do exist (Heitkötter, Hanschke, & Majchrzak, 2012). Nevertheless, this type of technological decision has its basis on the users that the developers want to reach. Moreover, the type of mobile the targeted users own may also tell something about that person's socio-economic and other contextual factors such as income levels or technical skills. When choosing the right platform or technological environment to build the application on, the question is not only on what the platform enables the developer to do. It also has implications for example in terms of application distribution and income generation (Holzer & Ondrus, 2011). What is noteworthy is that also these issues trace back to the targeted users in terms of what are the

best channels to reach the users or the best ways to generate income from the users. Different user groups prefer different distribution channels, and therefore also chosen distribution channels have a connection to the wider societal context.

The literature on design shows that design of applications, all the way from choosing the platform to deciding the right distribution channel can be linked back to the context where the usage of the applications is supposed to take place. Regarding technology development in organizations, the contextual factors have received more attention through the development of soft systems methodology (Checkland & Scholes, 1999), but targeting users that do not form part of the same organization is likely to differ from this type of systems development. Overall, as stated in the literature ICTs should not be seen as unique solutions to developmental problems, but forming part of a bigger picture regarding solutions (Walsham, 2017), in which contextual factors can be assumed of playing a major role. In principle, local application developers should be well positioned to be more aware of these factors, and incorporate those into the designs of the applications. Furthermore, they should be relatively free from path dependencies when they are situated in locations that have little or no tradition in software development. However, physical proximity does not necessarily imply awareness of all the contextual factors and path dependencies as well might be carried over in the form of technology. Understanding these issues can also help in understanding the arguments for the benefits of local technology production as well as further position technology's role in addressing societal challenges. In relation to application development, the issue becomes more complex because although the applications themselves are locally made, the technological resources behind them derive usually from elsewhere and have to be implemented by the developers to the local context.

2.4 Technology Implementation in Developing Countries

Contextual factors get often highlighted as the key determinants for successful transferral of technologies from one location to another. The topic is of special importance to many developing countries, where traditionally little technology has been produced, especially the more sophisticated kinds, and instead it has often been imported from somewhere else. To some extent this has recently changed in countries that have become part of global supply chains, and in the process these countries have also built production capacities to serve the home market (Foster & Azmeh, 2016a). Despite this, many of the least developed countries have had to rely on imported technologies, which have been designed elsewhere and often with other types of contexts in mind, which has led many technology driven development projects to fail (Heeks, 2002a). These projects have often been based on an assumption that a technology can

solve a particular challenge almost irrespective of the location, however studies have shown that contextual factors plays an important role in the implementation of these technologies.

Overall, a lot research has concentrated on the implementation stage of various technologies in the recipient locations, and the aim has been to discover the reasons why projects built around certain technology have either succeeded or failed (Bada, 2002; Braa & Hedberg, 2002; James, 2010; Miscione, 2007). These cases have constantly noted the importance of context, and for example the differences in design and implementation location have often been seen to be too stark to prevent successful implementation, which has also meant that the usage of these technologies has remained minimal. The underlying issues are usually linked to local norms, habits and traditions, which affect the usage of a technology and have not been taken into account in the design of the technology. Heeks (2002) for example refers to these as design-reality gaps, where differences in areas such as technology, management, skills or even intentions (Masiero 2016) have lead many of the ICT transfer related projects to fail.

2.4.1 Fit between Technology and Context

The key point arising from implementation studies is that technologies are never used in a vacuum but always in a particular context, and in order for a technology or a system to be successfully used, certain resources and skills have to be present and certain organizational and other arrangements made in the implementation location. On top of these, the local context places its own set of conditions for the technology implementation. Lack of designing context-appropriate technologies meeting local needs, not partnering with local organizations or building relationships with governments, or not adhering to socio-cultural norms or inviting participation from the targeted community are examples of contextual issues mentioned in the literature that decrease the chances of successful implementation (Toyama, 2011). In sum, the crucial factor for any technology to succeed is to understand the local conditions and requirements, and incorporate those into the design of the system or artefact (Walsham & Sahay, 2006).

In cases where technologies are brought in from somewhere else, one does not only transfer technologies but also values, rationalisations and meanings, which may not match with those of the implementation location (Miscione, 2007). As a result, certain amount of flexibility is often needed from the imported technologies in order to better match the local conditions and requirements. However, this is not entirely unproblematic, as this kind of thinking goes against processes like standardisation. Although standardisation can certainly simplify the overall functioning of a system across regions by increasing interoperability, it also imposes a more rigid implementation model that leaves less room for local adaptations. With standardisation,

technologies and systems lose some of their flexibility, which is likely to cause problems in implementation, especially if the implementation locations do not have similar standards in place (Braa & Hedberg, 2002).

As noted, one way to overcome those challenges is to increase the amount of flexibility, even at the expense of standardisation. However, also the society or organization that receives the technology should be able to adapt to an extent to the requirements put forward by the technology (Bada, 2002). Overall, contextual factors are often assumed as relatively static, but arguments have been made on their temporal and dynamic nature, leaving room for the possibility of adjusting local contexts in a manner that would better enable the functioning of the technology (Pettigrew, 1985). This can especially be the case if dealing with long term projects that have sufficient time for local context adaptations to occur (Njihia & Merali, 2013). Therefore, instead of only trying to adapt the technology for the realities of the implementation location, another approach for the technology implementation challenges is to prepare the contextual factors more suitable for the technology. Seen this way, technology or system implementation becomes a process of interaction between the technology and its intended usage, where both sides should have some capability to change and to adapt in order to better accommodate the requirements of the other (Macome, 2008).

Braa and Hedberg (2002) refer to this kind of adaptation process as cultivation, and see it as a way to solve challenges that originate from the incompatibility of intended objectives and the local contextual factors. Cultivation in its essence aims to tie the technology to the existing social systems so that it aligns itself according to the interests of its intended users. In the process the technology is hoped to gradually transform within the possibilities built into it so that is more in line with the social structures that exist in the implementation location. At the same time, new sets of standards are being cultivated into the existing social systems and local contexts to guarantee their compatibility with the technology. The expectation is that this way the overall context is more favourable for the implementation of the new technology. In the end, cultivation for the implemented technologies displays itself as a balancing act, where certain amount of flexibility should be included in the design processes to enable local adaptations, while at the same time keeping the technologies usable across regions (Rolland & Monteiro, 2002).

In sum, cultivation hopes to minimise the risk of a technology or system losing its meaning to its intended users by making the technologies and local contexts more compatible (Thompson, 2002). This does not mean however that the institutions, social contexts and historical trajectories that are in place in a particular location can simply be replaced by the ones that the transferred technology carries with. As noted above, without the active shaping of the local

conditions, the targeted users easily find themselves between two sets of values and rationales, those stemming from their local environment and culture and the ones that are incorporated into the technologies (D'Mello, 2005). By being able to find the possible incompatibilities already in the design or implementation phase, the problems caused by those can be avoided later on, as well as allow better understanding for the contextual factors that can be shaped in a way that increase the chances of success of a technology in the implementation location.

In addition to implementation, cultivation has been linked also another problem area of technology transfers, namely sustainability. Overall, implementation and sustainability are closely linked, but successful implementation does not necessarily guarantee continued usage of a technology or a system. Kimaro and Nhampossa (2005) see sustainability as an issue that can be understood through institutionalisation. Problems in sustainability stem from the inability of the transferred technologies to institutionalise themselves into the existing social patterns and habits. This can be due to a number of factors, ranging from inadequate infrastructure to lack in required skills. In relation to developmental projects, donors can also have an impact of making any implementation lack long-term support, since the experts that are provided for the projects are only appointed for a short duration of the process, or the implementation is done in a manner that does not incorporate the actual users, making it more unlikely for them to accept and use the system after the experts have left. All of these may have the unwanted consequence of inhibiting technologies becoming institutionalised in the implementation location and causing problems regarding sustainability.

2.4.2 Building Technologies Locally

Overall, the common factor for most of the problems regarding technology implementation and sustainability is the failure to successfully accommodate contextual factors of the implementation location. The proposed solutions emphasize the need for the technologies to be more adaptive and flexible. At the same time, there are claims that many of the problems the implemented systems and technologies aim to solve in developing countries go deeper than mere design strategies (Toyama, 2011). On top of this, technologies' capacity to adapt is often limited once that it has given its final form. This also means that integrating local factors and conditions after the design phase is easily problematic, especially in cases where the technology is derived from somewhere else.

This raises the question on the feasibility of local technology production. From a developmental perspective local production and consumption of ICT related goods and services is seen as a necessary condition for a developmental transformation to occur, especially in relation to the new, ICT driven areas of economy (D'Costa, 2006). However, the risk is that even with local

production, the benefits and the consumption of those goods and services remains within the already privileged groups within the local communities (Parayil, 2005), and as a result, the locally produced content presents the interests of those instead of the more marginalised groups. To change the situation requires deeper structures to be modified accordingly (D'Costa, 2003). Therefore in conjunction with promoting ICT usage and deployment in developing countries, the more traditional areas of development work, such as improvements in guaranteeing education, have to be addressed as well (D'Costa, 2006).

Despite these issues, local application development should in principle be better positioned to be aware and take into account the unique local factors that affect the successful incorporation of technology into the different areas of the society. This is not to say that local production would be problem free, as many of the challenges are relevant to all kinds of technologies, be that locally produced or not. The argument simply states that local technology producers might have an advantage in navigating around these challenges.

Overall, many of these challenges surface in the usage stage of the technologies. One area of literature that covers these challenges in usage particular well is the one on digital divide. It initially dealt mainly with issues regarding access to technology, and as such was based on the idea that lack of technology meant lack of development (James, 2007). However, recently the term digital divide has been understood in a deeper manner, and it has been broadened to take into account divides not only in relation to access but also in terms of skills and knowledge to use the technologies. Digital divide is of additional relevance for this thesis as the research topic area, local application development, implies that the divide has narrowed, or otherwise there would be no point in creating the applications for the local markets in the first place.

2.5 The Contextual Dimensions of Digital Divide

Digital divide describes the difference that exists between groups of people in relation to having and using digital tools and devices. In its crudest form digital divide refers only to being in possession or having some other way access to them, which means that the divide is seen as closed once people have gained access to the technological devices and tools in question, be they mobile telephones, laptops or Internet connection to name a few (Donner, 2016). On a global scale digital divide is often seen as existing between developed and developing countries, although there exists considerable variation between different societal groups within countries as well (Wareham, Levy, & Shi, 2004). As discussed above, the main reason for digital divide being seen as problematic is relatively simple one: ICTs are expected to have a positive impact for the local economy and for other areas of a society, and therefore it logically follows that if

people do not have access to these ICTs, these positive impacts cannot be obtained (Chigona et al., 2009).

2.5.1 The Different Definitions of Digital Divide

Seen in its simplest form, digital divide is mainly about finding ways to provide people access to ICTs, although already this view can be problematic for example in terms of which type of ICT one should promote. In the light of recent history, one particularly promising technology to close digital divide has been the mobile phone due to its relatively low cost and easy accessibility, which can be seen in the general high growth rates in ownership across the globe but especially in many developing countries (GSMA, 2017). The mobiles especially can be used across increasingly different usage cases, as even the more basic models become equipped with more sophisticated functionalities, and on the top end of mobile phones, smartphones tend to resemble more computers than phones. To increase the ownership in less developed regions calls have been made among other things for policy interventions to promote access even further, especially among the very poor who do not yet own a mobile (Napoli & Obar, 2014).

With the increasing ownership of technologies like the mobile phone, the digital divide between haves and have-nots has been generally seen as diminishing (Avgerou, Hayes, & Rovere, 2016). However, this applies only if digital divide is seen in relation to access and to one particular type of technology. The overall situation is more complex as research has pointed out that mere access might not be enough to close the digital divide. To start with, technologies like the mobile are not only about the device, but also about the devices' capacity to perform certain functions and to provide different services, which have to be built by someone. If those functions and services are not available, the usefulness of the devices themselves decreases, which further means that the same device can offer very different levels of utility depending on the services and functions available. As a result, digital divide, especially regarding the more sophisticated technologies, is not only about having access to the devices, but also what they enable their owners to do (Pedersen & Ling, 2002; Zainudeen & Ratnadiwakara, 2011).

Another level of complexity regarding digital divide is based on the dynamic nature of the term. Digital divide is often used indistinctively to cover several different technologies. As a result, if the divide can be closed regarding one type of technology, another one is likely to open as new technologies are developed and enter the market (Gunkel, 2003). Furthermore, there are no exact thresholds according to which it could be concluded when a digital divide regarding a particular technology ceases being a problem, or even how to decide which type of technological divide matters the most.

On the other hand, the provision of more technology is not always necessarily even considered positive, as technologies may also amplify the negative intentions and capacities. Toyama (2011) sees that technologies can serve to increase the differences between people belonging to different social classes and groups and favour those who are already in a privileged position, as they are better situated to acquire and use the technologies. Furthermore, the so-called high-achievers, who are often to be found among the privileged, are more likely to use these technologies in a beneficial manner, i.e. in the form of accessing material that enables self-improvement in relation to new skills and knowledge.

The argument made by Toyama is noteworthy not only because it points out the possible negative outcomes of increasing levels of technology, but also by discussing digital divide from a broader perspective. There the question is no longer only about access in material terms or even about the services that these devices can deliver, but also about users' ability to use these technologies in a meaningful way (Gunkel, 2003; Mubarak, 2015). Therefore, studies on the percentage of population having access to what device are not complete without understanding the contextual and other factors that impact usage of the devices (Al-Jaghoub & Westrup, 2009; Wareham et al., 2004). Gunkel (2003) argues that instead of a divide one should speak of digital divides, which have their basis in intersecting social, economic and technological differences. Van Dijk (2006) elaborates this by making a distinction between four different types of accesses: motivational, material, skills and usage access. The first one refers to the willingness of a person to acquire material that is needed to have access, such as a computer, whereas the second is more about whether the person actually can do so in relation to his or her income, gender or other demographic factor. The third type of access, skills, stresses the need of the people to have the required skills to use the relevant hardware and software, whereas the last access type, usage, basically requires the users to use the systems in order for the digital divide to shrink.

These four access types vary considerably between different technologies and it does not follow automatically that a technology that is easy to use would also be easy to acquire (James, 2007). As an example, basic mobile phones are often cheaper to acquire than for instance televisions, yet in relation to skills, televisions are possibly easier to operate than mobile phones. Also other factors come into play regarding the overall usability and utility. Some technologies and their utilities are heavily dependent on network effects, and in those cases limited access means also less overall utility. This is also similar if people do not have the skills to use the technology, as not having the skills can easily equal to not having access to the technology at all. Different technologies also enable different actions and contributions. Although mobile phones allow their users to do various tasks similar to the PC, the former's capabilities overall lack behind the latter, already due to issues like screen or keyboard size (Rice & Katz, 2003).

Thus, digital divide is not only about getting the technology into people's hands, but also making sure people have the relevant skills to use these technologies effectively (Mason & Hacker, 2003; Warschauer, 2003). This is not necessarily a problem, if the technology can bring benefits without people actually having to use it (James, 2006). In some cases this can happen, as lot of the functions get actually hidden from the users of a particular technology. However, technologies almost never function purely alone, but do have to be appropriated at least at some level by actual users. The way this appropriation occurs and the possibilities that particular technologies enable to their users are context dependent. Mere material access does not turn everyone into content producers, as socioeconomic and other factors affect the probabilities of this happening as well. The same applies for skills, where motivation and general interest are also needed for people to acquire the skills needed (Schradie, 2011).

As a whole, the literature shows that the seemingly simple initial definition of digital divide has become much more complex over time and gained more depth by incorporating the perspective of users into its definition (Warschauer, 2002). The argument further follows that the phenomenon that has been referred to as digital divide is not even a dichotomy, but instead more of a continuum, as there are several stages of in moving from one side of the divide to another (Gunkel, 2003). Skills or even motivations are not binary in their essence, something that one either has or does not have, but instead there are many levels to them. When combined with the amount of dimensions there are to measure the width of digital divide, closing the divide entirely along all the variables and aspects becomes basically impossible.

As a result, it might be more sensible to discuss the circumstances under which the divide can be considered as narrow enough, and determine the key dimensions that should be included in making the decision on this, without forgetting to discuss the implications of these definitions. Donner concentrates on these issues by looking into what happens, when in relation to physical access, the divide can be said to have been considerably narrowed (Donner, 2016). The technology he uses in his research is the internet or mobile internet in particular, and he makes the claim that regarding access the divide has been considerably narrowed as more people have the means to access Internet if they wish to do so. However, this does not mean that the access serves everybody at an equal manner, as usage or access to Internet is heavily conditioned by other factors that may easily go unnoticed. Internet connection of one person might be very different from another one's who is located somewhere else, as differences in data pricing, coverage, network speed and devices that are used to access Internet impact not only one's possibilities to connect, but also largely determine what one can do with the internet for example in relation to web browsing, sending mail or creating content. As noted above, this is not either a purely technical question since a lot depends also on the person's skills and

capabilities, levels of motivation and overall what one decides to do with the access. Furthermore, another factor that varies across locations is what is available for users. Local content might be abundant in some areas while almost inexistent in others. On top of all these factors, on a conceptual level it is not even entirely clear what constitutes Internet usage, as things like web surfing, chatting, application usage or email are seen as forming part of Internet usage in some cases and in others not, depending entirely on the definition (Donner, 2016; Gitau et al., 2010).

Despite the increased complexity of the term digital divide, it mostly still displays itself in the form of the technology and the users of the technology, but this type of dichotomy easily leaves out other relevant categories such as the content providers or the application developers that try to create applications for the devices. Furthermore, within a developing country context these developers are in an interesting position where they simultaneously have themselves leaped the digital divide (since if they did not have access to technology and some skills using it they could not create the content or applications), yet at the same time function in a context where many of their targeted users may at least to some extent still reside on the other side of the divide. In relation to access to technology, the assumption among developers most likely is that the users have access to the required technologies (otherwise it would not make sense to build the applications for the targeted users), yet regarding users' skills this is less certain. Acquiring these skills is not necessarily straightforward, as developing countries' capacity to provide the needed institutional arrangements for providing the skills may not always be entirely on par with the task, however the question is also then where do these skills come from.

2.5.2 Skills Transfer in Developing Countries

Having people with skills in ICT is seen as essential for countries to play a role in the global economy, as more industries and sectors of economies are dependent on ICTs and with that ICT-skilled employees (Lanvin & Kralik, 2009). In developing countries the discussion has built around where to get those skills, and overall many of these countries have been eager to promote ICT skills acquirement. However, it has not been always how to provide people these skills (Dunn, 2009; Tapia & Maldonado, 2009). Among others, NGOs and their programs that provide the participants certain ICT related skill sets have been suggested as part of the solution, yet the problem regarding many NGOs is their relatively small size and limited capacity to provide training for large masses (Mariscal, Botelho, & Gutierrez, 2009). Technologies themselves have been also envisioned to have a role in transferring skills to local communities and individuals by enabling work arrangements like telework (Dunn, 2009).

Another source for ICT skills development and skills transfer has been seen as coming in the form of software development outsourcing, where part or the whole development of software has been outsourced by companies to countries where labour costs tend to be lower. Outsourcing does not always have to be concentrated into large cities in the developing world (Schware, 2009), and one viable route for regions and countries to create a software development industry has been to become first a software outsourcing destination for foreign and in some cases also domestic companies (Lacity, Carmel, & Rottman, 2011). The idea is that foreign corporations enable skill transfers from their respective countries to the outsourcing location. This allows those skills to be used later on when establishing local companies and software development projects (Graham & Mann, 2013; Mann & Graham, 2016), although it has not been entirely clear how widely this actually occurs (Carmel, 2003; Carmel, Lacity, & Doty, 2014). Firstly, it has been noted that in order for multinational ICT companies to outsource operations into a particular location, that location has to be able to provide a pool of workers with sufficient skills before the multinationals enter and do any further training (Patibandla & Petersen, 2002). Claims have also been made that only low-skill tasks will be outsourced, and as a result the impact of outsourcing in relation to high-level skills transfer would be less significant (Gurumurthy & Singh, 2005). In addition, the low employee retention rates in some of the outsourcing locations may make the employers more hesitant to train their employees to be equipped with more advanced skills (Lacity et al., 2011).

There are also different types of outsourcing projects and initiatives, and for example in relation to so-called social outsourcing projects the role of public sector has also been highlighted to make sure that these projects can have more developmental as well as economic benefits (Madon & Sharanappa, 2013). Others have also noted the need for incorporating the environmental and developmental values into the business plans of responsible companies and the governments' role in this, and as a result, skills transfer among other things could become seen as part of the whole logic of a company's functioning (Markman, Russo, Lumpkin, Jennings, & Mair, 2016). Overall, cooperation between companies and local communities may potentially lead to outcomes that are beneficial both for the companies as well as for the local communities (Sandeep & Ravishankar, 2016). However, similar to the NGOs, it is not clear how large numbers of people could benefit from the skills transfer occurring in these kinds of impact sourcing projects. The approaches like outsourcing that might have a wider reach are also problematic as many of these countries have little history in being a target destination for any type of ICT outsourcing. Therefore the needed skills base has to be created either by educational institutions or via self-learning through other mediums (Dunn, 2009).

Start-ups have also been given a role in creating opportunities for people to acquire skills. In relation to new type of products and services entering the developing countries, the ability of existing companies to get these to the users has been questioned in the African context due to problems in agreeing on common standards and procedures. One solution for this has been to have start-ups to function as bridges to bigger companies that can in some ways facilitate the raise of the emergent market (Ozcan & Santos, 2015), and simultaneously get people to learn how to use the new products and services. This links back to another relevant question on skills transfer, namely how much of skills one should acquire. Even though there is always demand for high-end skills, it is also claimed that already a relatively low sets of skills are enough to boost the economies and enable people to grasp better employment opportunities in the less developed regions (Garrido, Badshah, & Coward, 2009; Walton, Putnam, Johnson, & Kolko, 2009), and the start-ups could be seen as having a role in this.

The issue for application development in respect to skills is therefore twofold. At the same time they need certain skills themselves to build the applications, but also their users should be skilled enough to use the applications. Already for this reason, digital divide is likely to affect application development process in particular ways, and the issue on skills is central for the success of the applications, both in terms of building the applications but also in using them.

2.5.3 The Digital Divide in Application Development

The common factor in almost every definition of digital divide regarding developing countries is that the technological artefact that forms the basis for the divide originates from somewhere else, i.e. outside the countries in question. This highlights the view of developing countries as receivers of technology but not as its producers, which is often the case especially in many of the countries in sub-Saharan Africa. However, local application development somewhat challenges this view as it includes a group that actively creates new digital solutions by producing applications locally.

Developers do have to accommodate their application development though in a manner that takes into account the technological level of the society, for example by looking what kind of technological devices are available for the targeted users and what kind of skills they possess. Despite this, local application development can be seen as having a role in bridging the divide by producing content and services that at least in principle have a better chance to match with the local needs and wants, as the creation of these digital technologies occurs within the society instead of being designed elsewhere and then transferred into these locations. Local technological production may also drive institutional changes that further narrow digital divides. One example of this is related to accessing the devices, where instead of buying a particular

device, alternative routes to gain access may exist, such as sharing (James, 2007). These kinds of factors are more likely to be taken into account in designing the applications if the designers are in the location and aware of the local conditions.

As noted, the closing of the divide is often seen as automatically beneficial, yet there are no clear answers on what exactly can be achieved from doing so, for example in terms of general developmental goals such as eradicating poverty (Gunkel, 2003). Furthermore, every location has its own particular context that affects how these issues can be solved, and as a result, within the discussion on digital divide there is a need to better understand what happens to a technology when it gets appropriated in a particular context. Furthermore, another aspect is how technology usage and digital divide are linked in cases where technologies can be actively shaped by the local communities to better serve their needs and wants.

Application developers occupy a position, where they are at the same time appropriators of technology as well as creators of it, although conditioned both by the tools and the capabilities of the targeted users. To some extent, the developers are at the same times on the both sides of the divide. On one hand, for the developers themselves the divide seems very much closed: applying van Dijk's four categories of digital divide, they can be argued as having the motivation, access to hardware as well as software, required skills and finally actively using the systems, and furthermore, if they were to be missing any of these, the applications simply would not be developed. However, for the success of the applications critical user masses are needed for a country or region to benefit from the technologies, and if the targeted users suffer from digital divide, be that in terms of access or skills or something else, the developers have to take this into account when building the applications.

This has also been one of the driving forces behind the enthusiasm to build especially mobile applications, as for many the mobile is the primary ICT device in many of the developing countries (Donner, 2016). For the developers, mobile phones, especially the more basic ones, mean additional constrains as content creation is to some extent limited in relation to the technological options available. This might imply also that the users' ability to reap full potential of the possibilities provided by ICTs, as people who only access Internet through their mobiles have been claimed likely being lower class internet citizens as they have only limited access to the Internet. The devices they possess can be used for retrieving information, but not capable of information creation like some other, more sophisticated devices are able to do (Napoli & Obar, 2014). This may also partly explain the findings that out of the new users of Internet fewer and fewer seem to produce content (Pimienta, 2008), and the risk has been the creation of so-called second-level digital divides, which refers to the divides in usage patterns even after access

to the devices has been achieved (Hargittai, 2002; Hargittai & Walejko, 2008). For a developer who aims to target these user groups, it also might mean further constraints in creating the applications, and as a result, the digital divide of the user group becomes to some extent her own.

However, at the same time the developers can narrow the divide especially in relation to local content. It has been noted that disparities in available content may become even more significant area of divide than technological access, since without relevant content the devices themselves are of little use (Napoli & Karppinen, 2013). Content created elsewhere might not be relevant for the local audience, and also it is easily created for devices that are not present at particular locations, or just fit poorly to the ones that are present in those locations, such as certain websites for mobile phones (Souter, 2011). In application development, many of the decisions concerning these issues are taken in the very beginning of the application development process, for example in terms of choosing what kind of application to create, or which technology or platform to use for developing the application. The latter is particularly evident for example regarding mobile applications, which are built on a particular operating system, and unlike web content that can be accessed from almost any device, these applications only run on devices equipped with that particular operating system (Wu, 2007; Zittrain, 2008).

In sum, digital divide in its broader definition is an area that significantly impacts application development in a developing country context, especially in relation to the usage of the applications. There is less research on how the local developers and tech start-ups deal with issues that can be traced back to the different notions of digital divide, as the discussion on digital divide has not so far looked at the impact of these divides in relation to technology production occurring in developing countries.

2.6 Conclusion

The literature review highlights the key areas in literature which are relevant in setting the basis for this research and identifying the research gaps that this research can contribute in. Table 2.1 highlights these areas and also describes their relevance for this research. Overall, in the literature ICTs have been seen as having a mainly positive impact on the development of developing regions and countries. Traditionally these technologies have been imported from outside the developing countries, which has raised questions about their capacity to address local needs and wants in a sustainable manner, casting doubts on the ability of these technologies to reap all the benefits that have been hoped to follow from their usage. The process of technology appropriation is context-dependent, and same technologies have had different outcomes in different locations and contexts. Overall, studies have noted that any kind

of technology appropriation has to be understood from a broader perspective, which in addition to the technological factors also needs to take into account the context where the technologies will be used. In many ways, producing technologies locally has been seen as having the possibility to avoid many of the pitfalls linked to importing technologies outside developing countries, because at least in theory local technologies should be better equipped to understand the necessary contextual factors affecting the usage of the technologies.

Application development provides one area of local technology production, and as such represents a certain paradigm shift regarding ICTs and developing countries as it enables these countries to move towards being technology producers instead of technology receivers In the literature, the research on software development occurring in developing countries has so far largely concentrated on mapping the macro-level factors that need to be in place in order for a thriving software development industry to be born.

Although important as such, these studies tend to pay less attention to local context in terms of understanding its deeper meanings for application development, and the assumption in them appears to be that as long as certain requirements are met, results will follow, no matter the location. Despite calls made by different scholars, less research has been made to understand the issues from the perspective of local start-ups and developers, and to study how different contextual factors display themselves and impact the application development process. This research aims to answer these calls by studying application development as a process that consists of three main stages, ideation, application building and application usage. During ideation, the ideas behind applications are gotten and the overall designing of the application takes place. As these ideas are derived locally, the likelihood of creating applications that are relevant for the local users should in theory be higher than in relation to applications created elsewhere. Furthermore, application development that takes place in a developing country context is less prone to be affected by existing technological and organisational path dependencies, leaving more room for truly innovative application creation. Whether and how this actually occurs is unclear, and to answer these is important in order to have a better picture of the possibilities of local application development and its capacity to address local challenges.

Regarding the actual building stage of the applications, it is noteworthy that despite allowing to design and build applications locally, most of the technological resources needed in application development, such as SDKs, APIs and the devices used to access the applications, have their origins in the developed countries. Therefore although the applications are produced locally, they still rely heavily on external technologies. The issues discussed in technology implementation studies shed light to the possible problem areas that may rise from this, such

as not being able to fit with the local contextual requirements. How far this is the case also in application development remains more of an open question.

In a similar manner, usage of the applications is dependent not only on the ability to address the needs and wants of the targeted users, but also on being able to overcome challenges related to digital divide, such as the users' having access and skills to use the applications. Therefore the final stage of application development, launch of the applications and their usage, is impacted by the context where this occurs. As noted, by producing technologies locally, the start-ups should be better positioned to address these issues. If they are capable of doing this, this research can provide examples on how to deal with these problems also in other, technology-related projects.

Overall, when studying this type of paradigm shift in technology production, it is important to have a deeper understanding of the relevant contextual factors and to study how contextual factors, in addition to certain widely used variables, affect application development process occurring in a developing country. Both technological and contextual factors are likely to be important, yet without grasping these fully we may be left wondering why certain areas thrive over others, or how application development in one place differs from another in terms of the challenges and opportunities that are faced by the key actors in the process, such as the start-ups developing the applications.

In order to understand these contextual dimensions of application development, it is necessary to define what is meant with context and technology as well as how they are expected to interact. This shall be done next in the next chapter that presents the conceptual framework of this research. First, the different definitions of context will be discussed, which will lead the research to adopt the concept of routines to have a more tangible way to operationalise and understand context. After this, the technological characteristics of the applications are presented, as they set the basis for application building by enabling and constraining options available for the developers. Finally, applications and context are given a common framework by using Kitchin's and Dodge's (2011) notions on code and space.

Area of Literature	Relevance for This Research	Examples of relevant literature	Areas of Contribution
Societal impact of ICTs	ICTs in general are seen as having a mainly positive impact on societies and to their development. However, contextual factors impact the ways that ICTs benefit the societies, and same technologies can have different results in different locations. Furthermore, just as the local context shapes the ways ICTs are appropriated, the ICTs also impact the societies. Traditionally many of these technologies originate from outside the developing countries, which has been problematic as the technologies have not been designed for locations like developing countries. Application development for local markets presents a type of paradigm shift, where countries that traditionally have been technology importers now have a role in technology production.	Chavula 2013; Heeks 2010; Heeks and Molla 2009; Toyama 2011;	The role of context in technology production for local markets in developing countries and the ability of application development to reach economic and other objectives.
Research on software development occurring in developing countries	The research on software development in developing countries has mostly been on the factors that enable the creation of software industries. Calls have been made for studying the topic from the perspective of local companies and to understand how contextual factors specific to particular location impact the application development process which can be roughly seen as consisting of three different stages, ideation, application building and usage.	Heeks 2006; Karippacheril et al. 2013; Xiao et al. 2013; Zainudeen and Ratnadiwakara 2011	Understanding application development in developing countries from the perspective of the start-ups and other relevant actors instead of as a sum of macro-level variables.
Ideation (design) of applications	As a result of the locality of the start-ups, they should be better positioned to be aware of the challenges, needs and wants of the local users and designing applications to address those. Furthermore, at least in principle there is increased possibility for truly innovative ideas and applications since the developers are not being restricted by existing technological or organisational path dependencies.	Dearden and Haider Rizvi 2015; Fuggetta and Di Nitto 2014; Little 2004; Nieusma 2004; Oosterlaken 2012;	The ways start-ups derive ideas for the applications and the factors affecting the ideation stage of application development.
Building the applications (technology implementation)	Although the developers design the applications themselves, the technological resources needed in the actual building stage of the application originate mostly from elsewhere. These resources leave room for the developers to decide what to build and how, the technological resources still need to be appropriated by the developers, which may share characteristics with the studies done on technology implementation in developing countries.	Braa et al. 2002; Heeks 2002; James 2010; Lacity, Carmel, and Rottman 2011; Miscione 2007; Walsham and Sahay 2006	How technological resources enable (and constrain) application development process occurring in a location that differ considerably from the ones where the resources originate.
Usage (digital divide)	Following the application building stage, the applications are launched and hopefully used by the targeted users. In addition to being able to address needs and wants of the targeted users, the likelihood of usage of applications is dependent also on the technological factors that impact usage such as digital divide. Divide can occur in terms of access but also skills to use the technologies, and the argument is that local developers might be better in overcoming the challenges related to digital divide.	Avgerou, Hayes, and Rovere 2016; Donner 2016; James 2007; Madon and Sharanappa 2013; Napoli and Obar 2014; Ozcan and Santos 2015; van Dijk 2006	The impact of digital divide to the usage of the applications and how start-ups overcome challenges related to the divide.

Table 2.1. Summary of the Key Areas in Literature for the Research

3 Conceptual Framework

> *"A 'happening in the world' is what needs to be understood. From time to time, and always in time, new forms emerge that catalyze existing actors, things, temporalities, or spatialities into new modes of existence, a new assemblage, one that makes things work in a different manner and produces and instantiates new capacities. A form/event makes many other things more or less suddenly conceivable."* (Rabinow, 2002, 180)

In order to establish the links between applications and the locations where they are created, it is necessary to have an understanding of the ways the two can be connected. The literature discussed in the previous chapter reviewed different perspectives on context and its importance in relation to technology appropriation, in particular identifying ways in which location and context can be important when developing new applications. However, in order to further study the role of context in application development, it has to be first understood what is meant with context and how to do research that is closely related to it. The existing definitions of context tend to be rather complex, although in the literature the concept has often been used rather loosely without defining it in great detail. When more attention has been paid for the concept's definition, the definition has depended on the type of research the scholar has been engaged with (Avgerou, 2010). For example, within the field of information systems research, context has been often understood as an organizational setting, whereas in development studies context has often been studied in a wider manner, incorporating different societal and cultural aspects into the concept.

The different definitions are not necessarily problematic as long as the concept is defined and made clear how context is understood in a particular research. This naturally applies for this research as well. Furthermore, due to its central role in this research it is not enough to just have a definition on what is meant by context overall, but the concept should be operationalised so that the objectives of this study can be met. In other words, context needs to be understood in a manner that makes the concept's usage feasible in studying its role in application development.

In addition to context, another key area within this research is occupied by the technologies that are needed to build the applications and also to use them. Therefore it is necessary to have an understanding on what kind of a technological artefact an application is, i.e. what are its main building blocks and how do they come about, and what is needed for its usage. Overall, an application can be seen as a particular type of digital technology, which at the same time is created by using other technologies but also requires additional technologies in order for it to function (Kallinikos, Aaltonen, & Marton, 2013). This highlights the connection between software and hardware: without the latter, the former cannot exist (yet not necessarily the other way around). By understanding applications as something that are coupled to other

technologies we can grasp what an application is and how it should be viewed within the confines of this research.

Overall, by defining and discussing what is meant by context and applications, the research has to further link the two concepts together. For this, a particular understanding of space and how it transcends technology is applied. Seen this way space as a concept moves away from the traditional and somewhat geographical definition to one that is constantly in the making. Within this making, technology and contextual factors have a role in shaping the change in space as it transfroms one one state to another. In sum, understanding space as something which is not static but instead dynamic and constantly changing allows this research to link its two main concepts together in a manner that helps to answer the research questions. This is important since as will be shown below, both context and application development cannot be understood either as rigid concepts but more as ones that constantly take new shapes and forms.

The order of the different sections in this chapter follows the sequences in application development process. The contextual factors are believed to have a significant impact in the ideations stage, and that is why they will be discussed first. After the ideation, the applications need to be developed by using the technological resources, which is also the reason the technological basis of the applications is discussed second. The third stage, launch of the applications and their usage is where the two, context and applications, come together, and become part of the concept of space as discussed in the third section of this chapter. As noted, these divisions are not exclusive and in reality the different stages overlap and even form loops. However, they are kept here separate with the aim of bringing clarity to the presentation of the conceptual framework of this research.

3.1 Context and Its Operationalisation

Overall, the term context is often used without defining exactly what is meant with it. The studies that have made more effort in trying to define the concept have often noted that this is not straightforward, and what is meant with context tends to vary. A standard definition from the Oxford dictionary mentions context as *"the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood"* (Oxford Dictionary). This basically states that context is more than just a physical location, and should be viewed as capturing different elements, which make something understandable to particular people. The word context comes from the Latin word *contextus*, which further points to the collective nature of context as *contextus* is a combination of words 'together' and 'to weave', which seems to refer to a common perception among group of people on how things are, and context as a concept appears to rise from the social fabric of a particular place or location. This does not

necessarily mean that the concept would have an entirely social basis, but more that the social functions as a mediator for the physical in addition to adding its own characteristics to the concept, which further increases the concept's complexity. It is not always clear which physical factors play a role in a particular context or how the social mediates the physical and also explains why the concept is subject to different interpretations. However, for a research that aims to use the concept of context, it also makes it increasingly important to define clearly what is meant with context in that particular research, and how the term should be understood and operationalised within it.

3.1.1 Defining Context

Some of the research on the contextual dimensions of technology has approached the topic by looking at how context is conceptualised in technologies that are labelled as 'context-aware'. Rahimian and Habibi (2008) for example see that in terms of context-aware applications, context-awareness goes beyond the mere physical location and tries to also capture other aspects of context. Borrowing from Chen and Kotz (2000), four categories are seen relevant in defining this type of context-awareness, which can be defined in terms of computing elements (such as connectivity, nearby resources), user (profile, location), physical (lighting, temperature) as well as time. Context in relation to context-aware technologies is therefore understood as a group of variables, each of which are seen as important in understanding and defining context. This is not surprising as such, since this is pretty much the only way a technology can understand context and be aware of it, i.e. through certain variables and the values given to them. However, it is also easy to see that this kind of definition of context leaves behind some of the complexity of the concept, and therefore runs the risk of offering a rather static view of context.

What is noteworthy is that when context is given a digital form, it also has implications for the understanding of context. Grudin (2001) argues that once this happens, the captured context is altered in profound ways. Similar to the notion of functional simplification (Kallinikos, Hasselbladh, & Marton, 2013), not all the relevant factors of context can be incorporated into the digital form, and as a result digitalization of context always leaves behind the parts of context that were not suitable for digitalization. In this way, the technological context is removed from its original framing, and once digitalized, it can appear anywhere in the planet, also in the yet distant future. Furthermore, aggregation or interpretation of context is different when performed by software or when done by biological, social and psychological processes. Grudin's notions are important in two different ways. First, it points out that technologies themselves can be seen as carriers of context, which can be transferred from one location to another with the technology. Second, it also draws attention to the characteristics of technology-mediated context, namely that it is a simplification as it leaves many relevant aspects of context outside.

In addition to context-aware technologies, context as a concept gets often mentioned in studies focusing on user experience. The problem there has been that the concept is not necessarily defined in any particular way, but instead the term gets often taken as self-explanatory. An example of this is a study by Ryan and Gonsalves (2005), who distinguish between objective and subjective measures to examine how a device fits a particular context. Objective measures refer to measurements such as how many mistakes the users make using the object or how quick they are in performing the given tasks, whereas subjective measures concentrate more on the user perceptions of the device, for example how easy they found a device to use. Context is then something inherent in the users, and their ability to perform certain functions is linked to the fit between their context compared to the one proposed by the technology. However, there is no definition of context in a precise manner as it is whatever the user thinks it is and displays itself in the usage of a particular object.

As these examples point out, context is often defined in a manner that serves the purposes of the research in question. This is not problematic as long as it is laid out clearly what is meant by context and how it is understood in terms of the objectives of the research. Some of the definitions above take for example a rather narrow understanding of context, but others use the concept in a broader fashion. In relation to application development, Dey et al. (2001) define context as *"any information that can be used to characterize the situation of entities (i.e. whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and applications themselves. Context is typically the location, identity, and state of people, groups, and computational and physical objects"*[6] (p. 106). In this definition, context includes multiple agents, both human and material, and becomes visible in the interactions that occur between the different locations, objects, groups and individuals, and within application development, special attention is given to the users of the applications. What is noteworthy that here context is no longer static, but becomes more of a process, where meaning is being constantly negotiated by the interactions between people and material. As a result, the definition also implies that contexts are dynamic and subject to change over time.

This is also something that other scholars have highlighted. Dilley (2002) for example talks about the social life of contexts, and by that he refers to context as something that is created by individuals and constantly being interpreted and constructed in the process. Context is an action of creating connections and disconnections between the relevant actors, objects and environments, and it is done by the person or persons who decide upon themselves what constitutes relevant setting in a particular case. In this sense context displays itself as fluid and

[6] The word *'state'* in the definition can either refer to a physical, social, emotional and informational state.

relational, subject to interpretations done by relevant actors and as a result, something that different individuals view in different ways.

However, it is important to remember that it does not automatically follow that contexts are easy to change. Drawing on Dilley, Hayes and Westrup (2012) argue that context could be better understood by using concepts of distal and proximal accounts, which they borrow from Cooper and Law (1995). Distal representations see entities as stable and discrete with clear boundaries. The description of these types of entities focuses on explaining how entities are in terms of structure, or what is it that these entities are expected to do. Proximal representations include the notion of time by concentrating on the processes that lead to certain distal representations. As noted, the distal forms of context are relatively stable, and although they do change over time but usually relatively slowly. This does not mean though that this dynamism of contexts should not be taken into account in research. One way to approach context in research is to identify contexts in their distal form, and then set to understand how they came to be by looking into the processes that lead to a particular view of context to occur. In sum, instead of narrowing context down to a particular type of setting or state where events take place, it is necessary to also take into account the dynamic and relational nature of context (Huen, 2009).

In information systems research context has been traditionally seen as a specific socio-institutional arena such as firm, industry or society, where technology gets diffused. The task has then been to understand how a particular setting impacts the appropriation or diffusion of technology, and the principal questions evolve around the enablers and barriers of the diffusion of technology like why the technology has or has not been used or diffused as widely as expected (Orlikowski & Iacono, 2001). Context matters because *"IT artefacts are always embedded in some time, place, discourse and community"* (p. 131). Avgerou (2010) further states that context plays an important role in explaining the social embeddedness of technology and its impact on socio-economic factors, and in order to better understand the role of context it is necessary to go deeper than merely viewing context in categorical terms such as organizations or local cultures.

This issue is further highlighted by Pettigrew's notion on the different levels of context (Pettigrew, 1985). He sees that context can be examined through analysing detailed description of events of change and problem-solving. The process should include both horizontal and vertical levels of analysis. With horizontal he refers to time, taking into account the interconnectivity between the past, present and the future of the researched phenomena. Vertical levels on the other hand point to the so-called inner and outer contexts, in other words he stresses the need to take into account intra-organizational factors as well as socio-economic

and political levels of analysis. This is in line with Avgerou and Madon (2004), who argue that even in a situated research it is not enough to concentrate only on the confines of the entity in question, but it is necessary to take a larger perspective of the institutional and other frames that also impact the conceptions of context. As a result, it is often not sufficient to pinpoint one particular aspect of context, as this also means that important contextual factors and aspects may end up being ignored. However, as noted above, what is important depends on the research area and focus as every research should adopt the definition of context according to its own needs, yet without drifting too far from the general understanding of the term.

In sum, there is no common universal and singular understanding what context is and how it should be defined. Context can be seen as something coming from the technology itself, from the interpretations of the actors involved, and also from larger institutional and societal settings. Furthermore, context is not static but keeps on changing with time and is influenced by changes in any related area, be that society, culture or individual interpretations. At the same time, the definition of context is dependent on the research, and it is up to the researcher to define the relevant understanding of context within each study.

Within this research, the problem with many of the definitions discussed above is that either they fall short on providing a broad enough understanding of context, or then they become too complex to be operationalised for the purposes of the research. The key aspect in defining the relevant context within this research is linked to the role of the developers, as the way context displays itself to the developers is vital in understanding how context impacts application development. To balance this and all the different aspects mentioned above, this research uses the concept of societal routines to explain the relevant context for the research. This does not mean that routines would be seen as equal to context, but more that societal routines reflect the existing contexts and this way function as proxies to the relevant context. The concept of societal routines also enables this research to operationalise the concept in a manner that allows the research to better explain the role of context in application development.

3.1.2 Routines as Proxies to Context

Studies on routines have mostly taken place within organizations like companies or company departments, where routines have been seen as forming an integral part of the everyday functioning of those organizations and workplaces (Cohendet & Simon, 2016; Feldman & Pentland, 2003; Parmigiani & Howard-Grenville, 2011). Overall, routines have been defined in different ways, but probably the most widely used is the definition from Feldman and Pentland (2003), who define routines as "*repetitive, recognizable patterns of interdependent actions, carried out by multiple actors*" (p. 95). The main point here is seeing routines as series of actions

that different actors repeatedly perform in a more or less standardized manner. The definition also states that those routines are recognizable, which implies that the actors, who partake in performing the routines, are aware of their existence.

The other definitions of routines largely point to similar directions. Prior to Feldman and Pentland, Cohen et al. (1996) for example saw routines as *"an executable capability for repeated performance in some context that has been learned by an organization in response to selective pressures"* (p. 684), whereas Nelson (1982) discussed routines by saying that *"our general term for all regular and predictable behavioral patterns of firms is 'routine'"* (p. 14). Both of these highlight the organizational aspect of routines. Routines are learned in an organization as an answer to the organizational necessities to perform certain functions and tasks, and in this sense become a regular event in the organization. The origin of routines is therefore in getting certain tasks done, and as a result, a pattern of action or a routine emerges to accomplish this.

Within routines and their performance some variation does occur and the way a routine is repeated and performed is not every time entirely similar. As a result, an organizational routine should be understood more as consisting of possible patterns that the performers of routines choose from to perform a particular routine. However, those patterns are being constrained, as well as enabled, by the different structures that are in place in an organization, be they organizational, physical, social or cognitive (Pentland & Rueter, 1994). These structures and also the general rules on which actions go together dictate largely how a routine is performed, and therefore the routine also maintains its recognisability and for most parts is repeated in the same form. What this means though is that routines and their performances are not mindless, but can be seen as *"effortful accomplishments"* (Pentland & Rueter, 1994), where the performing of a routine demands conscious actions from the actors involved, although it has also been suggested that routines are performed without much thinking (Pentland, 1995).

Overall, routines exist for the purpose of meeting particular objectives, and although not always performed exactly in a similar manner, they do possess a fair amount of stability. As a result, routines have been seen as causing inertia within organizations and being obstacles for change, as they have become to symbolize stability, control and overall preserving the existing status quo. However, as recent studies have shown that this is not necessarily always the case, since routines themselves tend to be constantly changing and can also function as tools to harness creativity (Feldman, Pentland, D'Adderio, & Lazaric, 2016). Sonenstein (2016) for example notes that when people engage with routines they also incorporate their own personal backgrounds into performing the routine, which may give rise to new ways of performing the routine.

In discussing routine change, Feldman and Pentland (2003) use the notions of ostensive and performative aspects of routines. The ostensive aspect of a routine tells what the routine is, and shapes the idea people have on the routine. This ostensive aspect relates to people's understandings of the routine and its purpose, and can be codified into items such as documents that describe how the routine is to be performed. The performative aspect of a routine on the other hand refers to performing the routine, where the relevant actors carry out the routine according to their understanding of it. These two aspects interact and shape the performance of the routines, which provides possibilities for variation, selection and retention of newly created patterns within routines. As a result, the ostensive and performative aspect also enable change in the existing routines within an organization. In addition to this, since performing any routine always leads to certain outcomes, those outcomes may function as sources for new ideas and routine improvements. For example if the performance of a routine does not produce the expected outcomes, the persons who participate and have a role in performing the routine may decide to change the routine. This can also occur if the routine performance leads to some specific problems that need to be solved. In sum, despite their inbuilt inertia, routines may also lead to new opportunities, which require adaptations to the existing routines or even the creation of new routines (Feldman 2000).

Routines tend to change with the organization they are found in as they are shaped by the different organizational structures in place in any organization. These structures can be seen as creating a particular context for the existing routines, and it is this context that shapes and determines many of the characteristics of routines (Orlikowski, 2000). This can also be noted in the discussions on the origins of routines in organizations. Routines have been seen for example as a way to reduce complexity and increase efficiency within an organization (Cohen & Bacdayan, 1994), provide a response to an organizational goal or stem from the environmental pressure (Cohen et al., 1996), or are results of the learning that occurs in organizations. Overall routines have been claimed to be a form of institutional legitimization as they signal a behaviour that conforms to generally established norms (Feldman, 2000). In relation to this research, the key point in all of this is that routines can be seen as responses to the contextual structures and objectives within an organization, and in this sense also rising from the organizational context and to some extent are highly context specific (Parmigiani & Howard-Grenville, 2011).

Therefore, it can be argued that although not the same as context, routines functions as mediators to the organizational context. They do so in a recognizable form, as mentioned in the definition by Feldman and Pentland (2003). Transferring routines from one organization to another can be difficult, as every organization has its own contextual characteristics, which also means that by looking at the routines that are in place in one particular organization, one can

also tell something about the overall context of the organization. Similar to context, routines are not stable but change with the organization and get altered or created as changes occur in their surrounding context. In other words, routines provide a proxy to the organizational context, and although they do not cover all the complexity of context within them, they do capture aspects such as the dynamic nature of context and also display a more multidimensional view of context than one given by few chosen variables. As a result, they also function as tools to study context in a manner that serves the purposes of this research.

The routines that this research is interested in are those that are performed in the society that surrounds the developers, and are therefore labelled as societal routines. Naturally, the scale is quite different when moving from an organizational to a societal level, but despite this, the main characteristics of routines remain largely the same. Also in societies, certain routines are being performed to reach certain outcomes in a repeated and recognizable manner, and they consists of interdependent patterns, which are being carried out by multiple actors. While true that a society can for example be claimed to be less unified, more diverse, and overall different than a company, especially in bigger organizations it cannot be given that everybody shares the same goals or has a clear understanding of the organizations' overall objectives. In sum, the routines that exist in a society share many similar characteristics than those that exist in an organization, and that those routines are conditioned and function as proxies to the context where they are being performed in.

What is more, societies have been seen as consisting of different organisations. In addition to organisations like firms, also families, clubs, and even states can be viewed as organisations (Ahrne, 1994). Although Ahrne (1994) does not state that societies are organisations as such, it can be claimed that if societies consists of organisations, every routine that is performed within them is pertinent to some particular organisation. As an example, shopping for groceries can be seen as a routine that is linked to the organisation of a family, and has the objective of providing food for the members of the family. Similar to routines in organisation science literature, this routine performance can include various actors, and not all of those have to belong to the organisation directly.

Examples of societal routines are various. In addition to the example of shopping given above, societal routine can be for example going from one place to another, such as going to work from home, or transferring money from one instance to another, such as paying the electricity bill. These societal routines should be seen as sets of interconnecting actions, which in most cases include various actors and are repeated frequently, and therefore fit the definition given by Feldman and Pentland (2003). Although routines may have similar outcome expectations across

different regions, the surrounding context impacts the ways these routines are carried out. A Londoner may wish to go from one side of the city to another just like a person living in Kampala, but the way this is done is likely to be quite different in terms of mode of transportation, price, routes, company, interaction with the people and even what to wear for the journey. This is also the way how these societal routines function as proxies to the overall context as discussed above.

In sum, societal routines share the characteristics mentioned in the definition of organisational routines. Furthermore, societies can be claimed of being organisations themselves or at least of consisting of different types of organisations, which all have their objectives and purposes. Due to these similarities, the concept of routines also functions on a societal level and within this research these routines are labelled as societal routines. When people perform these routines, the routine performances are shaped and impacted by the surrounding context, and as a result, concentrating on routines also allows the research to understand context in a manner that fits the purposes of this research.

3.1.3 Routine Attributes

In addition to routines, within information systems research the concept of habit has been used. The two have often been likened, as habits are seen as something that individuals keep on repeating though without much thought (Nelson, 1982). Habits have been especially linked to the usage of technology also outside the limits of organizations. Although largely missing from models such as technology adoption model (Venkatesh and Bala 2008), habits have been used to explain the continued use of IT and the factors that contribute to this usage (De Guinea and Markus 2009; Ligun and Hui 2013). Limayem et al. (Limayem, Hirt, & Cheung, 2007) define habits as *"the extent to which people tend to perform behaviours automatically because of learning"* (p. 705). Like routines, habits can be seen as products of particular contexts and as such, automated responses to different situations. Verplanken et al. (Verplanken, Aarts, & Van Knippenberg, 1997) see habits as *"learned sequences of acts that become automatic responses to specific situations, which may be functional in obtaining certain goals or end states"* (p. 540). They also make a distinction between habits and reasoned action, where the latter might have had a role in the beginning but as behaviour becomes repetitive, i.e. habitual, the reasons for particular habits fade into the background. Similar to habits, it does not automatically follow that any routine is perfect and that there would be no room for improvement. The views different actors hold on particular routines are subjective, and the estimations on the characteristics of these routines vary from one person to another. Therefore, routines can also be given particular attributes by the actors involved in viewing and performing them. This has further implications for the development of applications, as the applications themselves are

often based to the performance of the societal routines that exist in the context surrounding the application developers.

The success of a particular technology depends on its capability to solve a problem or to address a certain need or want that exist among the targeted users. Habits have been linked back to these through habit formation, as the needs and wants lead to goals, which on their part create an intention to reach that goal. Intentions, in turn, lead to behaviours, which if repeated long enough in a stable context will turn into habits (Ligun and Hui 2013). This applies also for routines, as they are performed to meet particular objectives and repeated on a constant basis.

However, as noted above, routines are not stable but keep on evolving and changing due to the changes that occur in their environment. Using the argument made by Hayes and Westrup (2012) regarding the distal and proximal accounts of context, routines can be viewed by looking at the structures they are based on, what the performance of the routines is expected to achieve and where they might be heading. Especially the last part implies that there is an element of change involved in the routines. Basically if routines change, then the question also raises why there is a need for a change to occur. One answer to this is that either the routine could be performed better to meet the expected outcomes, or that it is no longer capable of delivering the outcomes that are expected from it overall. As a result, the mere existence or establishment of routines does not mean that those routines would be ideal, and routines may fail to incorporate relevant aspects. Wilensky (1967) has noted how routines may fail in transferring information, or for example in the case of Pearl Harbor, Wohlsetter (1962) stated that information was available but went unnoticed since that information did not flow within the organization as expected by the existing organizational routine. Routines may therefore be subject to evaluation, which is based on an estimation that there are areas where a particular routine does not deliver the wanted outcomes and therefore needs to be restructured (Feldman & Rafaeli, 2002).

In organization studies, this type of re-evaluation of a routine may also take place for example when introducing new technology to an organization, and overall learning and improvement of routines can also occur informally from performing those routines, although the learning is likely to stop and routines become more stable once new routines have been established (Robey, Boudreau, & Rose, 2000). In order for the re-evaluation to occur, the argument this research makes is that the routines need to be given certain attributes[7], which basically describe a particular routine and its performance in terms of its characteristics and quality. For example, if

[7] The word 'attribute' (noun) here is understood as *"a quality or characteristic that […] something has"* (Cambridge Dictionary 2017)

a routine needs to be changed, the existing routine can for example be described by giving it attributes such as 'ineffective', 'costly', or 'does not meet the expected outcomes'. Similarly, if the routine works well, the attributes are likely to be more positive. Also more neutral attributes can be attached to particular routines such as 'stable' or 'performed by department x'. The point here is that in order for a routine to be changed, there has to be some way of describing and evaluating it, otherwise there would be no basis for changing or any implication how it should be changed. These attributes are largely subjective, and their interpretation is dependent on the person doing the evaluation and also the angle that the routines are being looked upon.

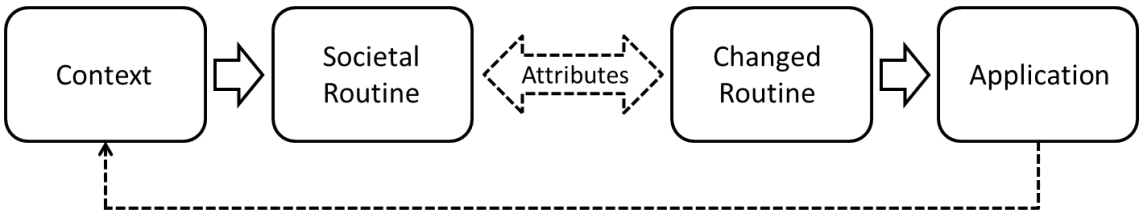

Figure 3.1 The Connection between Context and Applications

What this means is that routines themselves can be assessed by the people viewing or performing them, and as they are linked to the surrounding context, change in routines implies changes in the context and vice versa. Using the notions by Feldman and Pentland (2003) on ostensive and performative aspect of routines, if a change in a routine or a new routine is proposed, it also needs to be codified into items that describe how the routine is to be performed. This codification is based on the understanding on the existing routine held by the people who propose the new routine. The argument of this research is that this ostensive aspect of a routine is manifested in an application, which functions as a model for the targeted users on how the developers would like them to perform a particular routine. This also implies that the developers are trying to alter an existing routine on the basis of some, most likely negative, attribute in an existing routine, which they think their application is able to fix. Therefore, the technology becomes as a mediator for existing routines but at the same time also aims to change those routines. Figure 3.1 displays this process, and shows how this process creates a particular type of loop between context and applications. However, to better see how this occurs, it is necessary to have a deeper understanding of the technological properties of applications.

3.2 Applications as Technological Artefacts

> *"Applications consist of suites of IT capabilities. They are developed to meet a set of specified user needs within a select set of communities. They can grow amazingly complex in terms of effort and scope, but despite this, they still can be viewed as applications, if governed by a set of specifications through which their design scope remains bounded. An application is a priori determined by choice of design context, user groups and functional goals"* (Hanseth and Lyytinen, 2010, 2).

Adapting from Farrell and Weiser (2003) layered structure model as well as from Benkler's (2006) three-layered model, Yoo et al. (2010) see that digital technology consists of four different layers of architecture: devices, networks, services and contents. The device layer is where the basic hardware resides, whereas the network layer captures the hardware linked to transporting data (e.g. cables, radio spectrum) as well as takes care of the rules governing the data transmission (network standards e.g. TCP/IP). For the applications, the service layer is crucial as that is where the application functionalities are run. The layer on top, i.e. content layer, refers more to capturing and presenting data in forms such as text and images. One layer is built on top of another, thus creating a hierarchical order between different layers. The layers are seen as loosely coupled, which basically refers to the point that the different layers and the modules within them can be dealt separately and connected to other layers through clearly-defined interfaces.

As a result, Yoo et al. (2010) argue that this has led to a new type of product architecture, which they label layered modular architecture. They see it as a mix of modular architecture and layered architecture, where the former is about products that are compounded of loosely coupled components through interfaces, and the latter about digital technologies being embedded into physical products and thus bringing in capabilities in the form of software. The application developers find themselves on the service layer, which is higher up in the layer stack, and rely on the layers below in their functioning as those provide the upper layers resources needed for the applications to run.

In a similar manner, the layered modular architecture of digital technologies has enabled the creation of application platforms. These platforms have allowed external contributors such as third-party application developers to produce content and applications to the technological devices (Eaton, Elaluf-Calderwood, Sørensen, & Yoo, 2015). The result has been hundreds of thousands of applications varying in complexity and purpose that have been created by both individual and companies of all sizes to different application platforms (Tiwana, Konsynski, & Bush, 2010). Overall, these platforms combined with the applications have given birth to application ecosystems, and at the same time lead to significant changes in the innovation processes of many companies. Among other things, the boundaries of firms' innovation processes have become blurred, as instead of doing everything in-house, the layered modular architecture, and within them the platforms, have enabled the participation of multiple actors, some of which having very limited connection and relation to the firm in question (Maurer & Tiwana, 2012). As noted, more and more innovation processes no longer take place within the companies themselves, but are taken care of by a sometimes large number of relatively

independent actors, which have no formal linkage to the platform owners apart from the technologically mediated user agreements, technological resources and distribution channels.

These structural changes have led to a situation where the innovation practices have become highly distributed as well as combinatorial (Yoo, Boland, Lyytinen, & Majchrzak, 2012). Distributedness can be seen in the geographical dispersion of digital innovation activities, in the heterogenic nature of actors and knowledge resources, and in the convergence of machines through the linkages offered by digital technologies. In other words, this type of innovation can occur in different locations, be implemented by a wide number of actors in a manner that combines several technologies. Digital technologies have also become more combinatorial, as they enable connections to resources that can be seen as external also to the platform owners. These connections are made through the usage of application programming interfaces (APIs). Overall, these APIs and also software development kits (SDKs) that are used to build the applications have enabled software applications to use other digital artefacts as building blocks for their own applications. Developers draw resources from these various sources and recombine them in different ways. From a technological perspective these resources can be seen as ubiquitous since as long as one has access to some hardware and an Internet connection, they are available everywhere and at any time.

However, the usage of these resources takes place in a certain geographical area or social environment, and the applications that are created may base their reasoning and purpose on the local necessities of what is needed or wanted from an application. Therefore the creation of applications can be seen as one where globally available resources are put into use in a local setting. What is noteworthy is that the digitalisation of these resources is likely to impact the resources themselves. As Bailey et al. (2011) note in their research on virtual teams, digitalization leads to a certain type of de-contextualization. When items become digitalized they get detached from their form, and in the process lose some of their physical characteristics. Following this thought, the same happens to the resources that application developers use since these resources get removed from their origins and linked to a more global context, at least in the sense of being transferred across the world to different usages and purposes.

This phenomenon is linked to the concept of generativity, of which SDKs and APIs can be seen as prime examples. Zittrain (2008) defines generativity as "*a systems capacity to produce unanticipated change through unfiltered contributions from broad and varied audiences"* (p.70). Tilson et al. (2010) refer to a slightly different definition which states that generativity *"denotes the ability of any self-contained system to create, generate, or produce a new output, structure, or behavior without any input from the originator of the system"* (p. 750). Especially in relation

to the latter, the SDKs provide software developers a certain set of resources and tools, out of which the developers then choose the ones which are most appropriate for their own purposes in terms of building an application. What is more, the loose couplings that exist between different software components and modules enable the usage of additional resources via APIs, which do not originate directly from the SDKs. Furthermore, many of the applications themselves possess capabilities that are normally assigned to platforms, for example in the form of APIs that other applications can then connect to and draw resources on. Therefore the level of generativity within application development is likely to go even beyond the resources provided by the platform owners in the form of SDKs, as other actors that are external to the SDK providers also offer additional sets of resources that can be used.

As a result, applications can be seen as digital artefacts, which rely entirely on other technologies both in terms of their development as well as overall functioning (Kallinikos & Mariátegui, 2011; Yoo et al., 2012). They cannot exist as standalone items, but are dependent on the other technologies to provide them with the resources needed. As an example of this, an application called Ushahidi, which is a reporting tool that enables its users to report different incidents, is built upon and is dependent for its functioning on other technologies and applications. The Wikipedia entry of the application describes the following:

> *"The Ushahidi platform is built on the Kohana web framework, a fork of the CodeIgniter framework. It includes support for Nexmo wholesale SMS API and Clickatell SMS Gateway. Furthermore, the official Ushahidi-hosted websites use the commercial service. Ushahidi provides the option of using OpenStreetMap maps in its user interface, but requires the Google Maps API for geocoding. Ushahidi is often set up using a local SMS gateway created by a local FrontlineSMS set-up."* (Wikipedia, 2014)

Based on the description above, it can be counted that Ushahidi bases its functioning on six different external technologies[8]. The complexity and interconnectedness between various actors can be seen as being characteristic to information infrastructures, which Hanseth et al. (2010) describe as *"a shared, open (and unbounded), heterogeneous and evolving socio-technical system (which we call installed base) consisting of a set of IT capabilities and their user, operations and design communities"* (p. 4). Although applications in most cases fall short of being considered as an infrastructure in itself, it can be seen how an application can become part of these infrastructures by being in the networks that consist of other applications and various types of actors. However, the level of connectedness an application has to the external resources and other applications depends on what the application aims to achieve.

[8] And most likely to other additional technologies as well that are not mentioned in the description.

Therefore from the perspective of the developer it becomes essential to know which resources to use and what type of implications those resources carry with them. While offering different kinds of possibilities, each resource also places its own constraints on the application development process. Most clearly this can be seen for example in the form of choosing which platform to build the application on, as platform owners are able to control the resources that are available for developers and also in some cases decide whether the developers can use the platform to distribute their applications (Eaton et al., 2015). What is more, the resources that are used and the platforms on top of which the applications are built are often based on certain assumptions and standards that the application developer has to agree on, as otherwise it would be next to impossible to use those resources and platforms. However, this is not to say that the platforms and resources decide deterministically what kind of applications will be built. It merely points out that if digital technology is to carry within it contexts of its own, and by so doing affect the development of applications, this will happen in the form of resources that the technology offers. Therefore it is important to understand what kinds of resources are talked about, and to see how these resources function both as enablers as well as constraints in the creation of mobile applications.

3.2.1 Technological Resources as Affordances

From a more theoretical perspective the technological tools and resources that are used in the application process can be viewed as providing developers a set of affordances. The theoretical concept of affordances was first presented by Gibson (1977) within the field of ecological psychology, where he used the term to describe what environment affords to the animals, in other words what it allows them to do but also not to do. Affordances include all the possible actions that can be performed within the boundaries that the environment sets, and those affordances exist whether or not recognized by the individual. According to Gibson, the affordance is however relative to the actor, since it depends on the actor's qualities and capacities to make use of the affordances provided.

Norman (1999) used Gibson's notion on relative affordances to talk about the perceived affordances and their implications for design. Good design makes visible the affordances that it provides, and allows the user to quickly spot how to use the object in question. Gaver (1991) described this by categorizing affordances in terms of two dimensions, perception and existence, and based on them the affordance could be seen in one of four ways: a perceptible affordance was both real and perceived, hidden affordance real but not perceived, false affordance perceived but not real, and finally a rejected affordance did not exist since it was neither perceived nor real. In general there have been views that link the affordances more to the interpretation of technology or object on part of the individual engaging with it (Hutchby,

2001; Markus & Silver, 2008). As Leonardi (2013) states, materiality does exist independent of the people, but the affordances and constraints it provides on the other hand do not. In relation to application development, the developers have to be able to spot the possibilities and the constraints present in a particular technology, but in addition they must be in possession of certain skills that are needed to actualise the affordances in accordance with the objectives that are set for the application. It is also important to keep in mind that the technology itself also functions as a constraint by simply not allowing certain things to be done. In this sense it is the technology that places the ultimate limits on what can be done, not only the individual who is unable to spot them.

This can also be explained by using Norman's (1999) distinction of three different types of constraints, namely physical, logical and cultural. Physical constraints are connected to Gibson's original definition of affordances stating that it is not possibly to perform action if the physical world does not allow it. Norman's second and third constraint type make the term affordance more subjective, since the logical and cultural constraints depend on the individual's capacity to reason and also of his or her cultural background. In application development, the resources offered by the platform and the other technological sources via APIs can be seen as Norman's first type of constraints. Although there are different resources to choose from and those can also be combined in many ways, the developers have to take these resources as they come and can for example tweak them only within the limits allowed by the owners of the resources. Logical and cultural constraints on the other hand have their basis in the social, as they leave more room for negotiation on what is meant with logical or what the culture dictates.

Although the physical constraints decide the ultimate limits for the application, by being able to choose from different affordances, such as on which platform to build the application, the developer can also bypass some of the constraints that might be present in particular technologies. Furthermore, as shown above, an application can be viewed as a bundle of different technologies, and thus it can also be the case that some technologies enable overriding the constraints placed by others. Even then, it does not follow that anything can be built; there is always an ultimate limit set by the technologies available.

However, technological affordances and therefore also constraints can be seen as socially constructed due to their temporal nature. Corral et al. (2014) talk about evolving and inherent constraints, where former refers to constraints that are current but will be solved in the future due to the developments in technology, resources, signal coverage or other areas. Inherent constraints on the other hand are intrinsic to platforms and targeted devices as they have been implemented to the design, and tend to be more stable due to the fact that they have adopted

a space as key characteristics of the execution environment. In other words, when the decisions are made on which platform to use or which devices to target, it can be more difficult to avoid these types of constraints or to bypass them.

Zammuto et al. (2007) argue that technological objects and the affordances they provide should also be viewed as social objects in the sense that they affect how the social is organized. If the social context adapts to the technology, the factors that were initially perceived as constraints may cease being ones at a later stage. This highlights the temporal nature to the concept of affordances, which in the case of application developers mean that the possibilities and constraints of different technological resources may alter over time both in terms of the technology and in the environment where they are used.

3.2.2 Application Developers as Affordance Choosers

Overall, whether the affordances are located in the technology or in the user's ability to perceive the affordances correctly, the central figures to capture these different types of affordances are the application developers. They face the task of transferring the design context, relevant user groups and functional goals into the actual applications by using the technological resources and actualising the affordances they provide. However, it is noteworthy that the constraints that come with the technological affordances are not always based on technological factors alone, and that the actors that are in control of the technological resources used by the developers may also put in place constraints that have their basis in regulatory and organizational aspects and not on the technological possibilities per se. In this sense the providers of technological resources govern the developers through the means of technology in a manner presented by Lessig (2006), who sees technology as a regulatory force similar to legal frameworks. This can be seen especially in relation to the discussion on boundary resources, which share many similarities with the notion of technological affordances. Ghazawneh et al. (2013) define boundary resources as *"the software tools and regulations that serve as the interface for the arm's-length relationship between the platform owner and the application developer"* (p. 174). Boundary resources therefore function as toolkits that enable the transfer of application creation capabilities from the platform owners to the third-party developers (von Hippel & Katz, 2002), but also capture the notion of control or securing[9] as sometimes referred to in the literature on boundary resources (Ghazawneh & Henfridsson, 2013). This type of control can be exercised via developer agreements, but also in terms of technology by not providing certain resources to the application developers or not allowing the application to be distributed through the channels controlled by the platform owner. An often mentioned case is the Apple Store,

[9] The term securing describes the aims from the part of the platform owners to prevent the creation of applications that may infringe the platform (Ghazawneh & Henfridsson, 2013).

which functions as an only official distribution channel for applications created for Apple products such as the iPhone (Eaton et al., 2015; Ghazawneh & Henfridsson, 2013). This further highlights the need from the part of the developers to choose the tools and resources that best match their objectives that they have put in place for the applications they create.

In conclusion, the technological basis of applications displays itself in the form of technological artefacts, resources and tools that the application is dependent on in its development phase but also later on in its usage. Applications can be seen as combinations of different technologies, which are chosen from a large pool of alternatives. Each technology comes with its own set of affordances and constraints, which are partly based on the technologies themselves but also dependent on the intention of the owners of these resources as well. These resources are also the ones that place the ultimate limits on what can or cannot be done by the developers, although these constraints and affordances are also likely to change over time as technologies develop and resource owners' interests change. After this however it is up to the developer to choose the resources that best match his or her intentions and objectives set for the application. This however is not a simple process, as it depends on the developers' capability to spot the most suitable technological affordances and also on the personal skills to actualise those resources. Furthermore, cultural and logical aspects may also function as constraints as noted by Norman (1999).

Figure 3.2 portrays how application developers choose the relevant technological resources, be they in the form of APIs or platforms on top of which the applications are built, and further turn them into applications. These platforms are connected to particular devices, and choosing a particular platform also has implications for what kinds of devices it can run in. The developer functions as a mediator between the technological resources and affordances they provide, and the application that is built by using those resources.

Overall, it is these sets of technological affordances in the form of applications that get meshed with the societal routines discussed above. This occurs latest in the stage when applications are fed back to their targeted users, who then either start using them or reject them. In order to theorize this connection between societal routines and technological affordances a particular view of space and the role of code in it is adopted, which shall be discussed next.

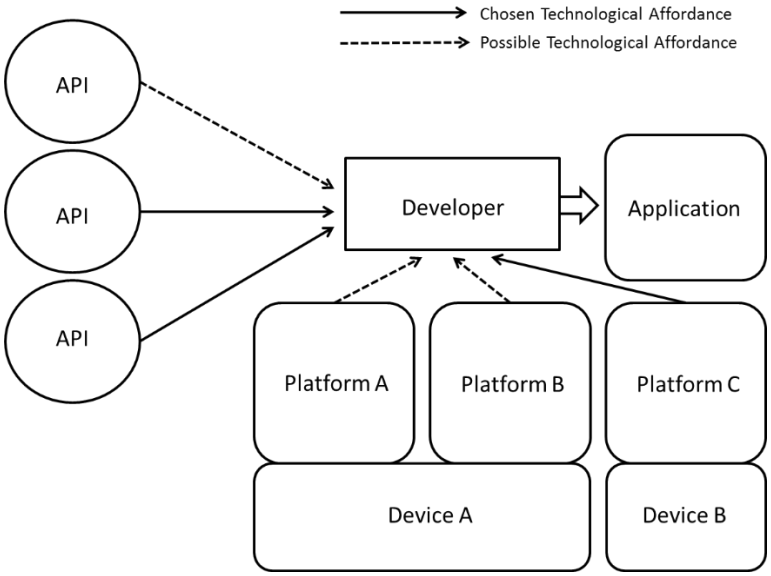

Figure 3.2. The Technological Basis of an Application

3.3 Code and Space: The Relationship between Technology and Context

Code understood as rules and instructions of software takes various forms and can be used in different technological settings and items. Although the syntax of code changes, the main purpose of code remains largely the same in terms of giving instructions to the hardware layers on what to do and how, no matter the connection or device it is used in. As the amount of digital technology increases in importance and becomes more prevalent in the everyday lives of people, code occupies an ever larger place in managing events and situations that occur in different places of the world.

Dodge and Kitchin (2005) see that code or software can be present in people's everyday lives in four different ways, which they refer to as coded objects, coded infrastructures, coded processes and coded assemblages. Coded objects are singular objects that depend on code in their functioning, or then items like data storages that cannot be accessed without code. These objects differ in sophistication and complexity, and examples of them range from items like washing machines, which have code installed to them to add functionalities to their usage, to isolated devices such as PCs, credit cards and USB sticks. Coded objects can be further divided into groups based on how much of their functioning is based on code and their overall relation to other objects and contextual factors. Peripherally coded objects have some code in them, but are not dependent on it to perform their primary task. Hard coded objects, or codejects as Dodge and Kitchin call them, are dependent on code but are not programmable and thus not very interactive in this sense. Unitary codejects on the other hand are, and may possess some capacity to sense and react to the world around them. Logjects differ from unitary codejects in

that they record their actions in logs that are stored and used in the future, and may possibly rely on other networks in their functioning, possess high levels of interactivity and multifunctionality. As can be seen, the scale moves from objects where code has only a secondary role to objects which basically rely entirely on code and require connections to other networks in order to function (Kitchin & Dodge, 2011).

The second category, coded infrastructures, refers to networks that link coded objects together as well as to the infrastructures themselves that are monitored and regulated by using software, i.e. code. Code forms an important part of these structures that consist among others of physical, digital and electronic components. Coded infrastructures are for example computing, utility, communication and broadcast networks, but for instance cars can also be seen as coded infrastructures which in addition to physical, mechanical and electronic components, have also code-run systems like the one managing the functioning of the engine. Code mediates the interconnections between the different components and therefore also controls them. Coded processes, the third category of Dodge and Kitchin, are the transactions and flows of digital data across the coded infrastructures. Coded processes are less about code that controls the process but more about the transfer of information in the form of digital data. The data is stored in databases that can be accessed across distances and linked to other databases, and overall are used regarding various aspects of people's lives, ranging from health and taxation to banking service usage. A typical example of a coded process is the usage of an ATM, where data transfers for instance to verify users, updating bank accounts and running security checks are done to perform transactions. Coded processes become visible in the form of bills, licences and receipts as well as in the form of coded objects like bank cards and tickets that are used to access coded processes.

The last category of code, coded assemblages, takes place when several coded infrastructures and processes get connected and are needed to operate in connection for a particular environment to be produced (Dodge & Kitchin, 2005). The example of coded assemblages Dodge and Kitchin (2004) give is air travel, where coded infrastructures and coded processes like billing, ticketing check-in security, immigration and traffic control come together.

Overall, these four elements or categories of code make the prevalence of code in people's daily routines relatively clear. Code has a role to play in enabling and controlling many of the daily activities that occur in societies, especially in the more modern ones, and in many instances people are dependent on code and its functioning. As a result, code can be seen as forming part of the different routine performances, where the main point of interest is not in the relatively static nature of code being implemented in different objects and infrastructures, but more how

code is involved in the functioning of the societies. Dodge and Kitchin (2005) discuss this by using two separate but in many ways linked concepts, technicity and transduction.

Technicity refers to Mackenzie's (2001) notion on technologies' role in mediating, supplementing and augmenting people's collective lives and how technologies in interaction with people have become essential in terms of enabling the performance of certain behaviours and actions (Dodge & Kitchin, 2005). Technicity as a concept is more than just the technology or technological object in question, but instead refers to the overall practice which has evolved around technologies or the usage of those technologies in performing a particular practice. In addition to the technology, technicity therefore also captures the human operators and the context in which the technology enabled operations occur, and by doing so, describes the role technologies perform in particular actions (Star & Ruhleder, 1996). As a result, technicity of an action is a matter of degree, where the levels of technicity of different activities and events vary.

Technological objects and technologies can be said to possess certain amounts of technicity, based on their ability to be used across different contexts and actions. As an example, software or code in general is seen to have a high level of technicity as it enables people to perform various practices and functions across various contexts, and to some extent, in a relatively similar manner. However, at the same time technicity does not mean that a routine that is subject to the functioning of code becomes universal or deterministic, as technicity of any routine is dependent on the contextual factors that are in place in the location of its enactment (Dodge & Kitchin, 2005). As technicity becomes embedded into a particular context it is said that it is moving from an abstract state to one of concreteness. In other words, a high level of technicity means that a technological object gets concretized to a context by becoming more interconnected to the different actors and realities that exist in the location of its usage (Mackenzie, 2003).

Transductions can be seen as processes where one state of things shifts to another state through repetitive or transformative individuations. Individuations are individual instances that occur in a sequential manner within these shifts from one sate to another, and as such from the sub-particles of these shifts in different states and when put together form a series of actions and events that enable the transduction, or the process of moving from one state to another, to occur. These transductions create relations between items, events or areas that have not been necessarily interconnected before (Mackenzie, 2003). An example of transduction could be for example shopping a product, where certain individuations like going to a store, choosing a product and paying it are performed, and as a result, there is a shift in one set of affairs to another for example regarding ownership.

Technicity becomes visible in transductions. A transduction can be said to capture certain amount of technicity, and it may therefore be dependent, among other things, on a particular technology to function, or otherwise it cannot occur or it will occur differently (Dodge & Kitchin, 2005). An example of this could be performing a financial transaction by using online banking. The transduction, financial transaction, is performed by using the technicity of online banking, and if the technology in question does not work, the transduction will not occur, or then the person has to go to a branch of the bank that may perform the transaction without resorting to that particular technology.

The transduction that occurs is dependent on the individuations that have taken place within it, and in this sense different transductions follow each other in a continuous flow (Mackenzie, 2005). In line with Sismondo (1993), Mackenzie negates the static nature of context and sees it more as a constant flow of individuations and transductions. Transductions themselves do not come out of nowhere, but instead are based on the transductions and individuations that have occurred so far, meaning that contextual factors overall condition what can be done and with that, what is done. Transductions can also be understood as responses to relational problems, or in other words as actions that follow in order to get something done or performed. This can be the need to do a financial transaction as mentioned above or for example to move from one side of a town to another. In both cases the so-called problem acts as a force that pushes events or individuations and with that, transductions, into motion (Dodge & Kitchin, 2005).

When comparing technicity and transductions, technicity is more about the process of technologizing particular events and actions that people perform or about the role technology plays in those situations. Transduction on the other hand concentrates on the performance of those actions in general, or how they come about and occur. Overall, the essence of technicity and transduction is in focusing how situations end events unfold and technologies' role in this, rather than merely looking at how a situation or an event is at a particular moment (Mackenzie, 2006).

Transduction, and with it, technicity, have their basis in the ontogenetic view of space. Instead of being something that exists, space seen ontogenetically implies that space is something that is constantly in the making. More than a relatively static concept, space should be seen as a flow of events, habits, situations and others, which actively shape what kinds of spaces are being created in a particular moment. Seen this way space has a clear temporal character, and it could be further argued that a particular space only exists a brief moment. By seeing space as a flow does not imply that things like institutions, contexts, cultures or norms would not have any say in the way spaces are being formed. It is more that they too are dynamic and interact with other

factors, and together form particular type of ontogenetic spaces through transductions that occur within those spaces (Kitchin & Dodge, 2011).

3.3.1 Routines as Transductions

The concepts of technicity and transduction therefore enable a study of context from the basis of routines, which individuals keep on repeating in particular ways and in connection with other relevant actors, events, objects and contexts. A routine in this sense can be seen as a type of transduction, which consists of individual steps and is often repeated in a more or less similar fashion. Furthermore, just as transductions can be seen as solutions to relational problems, routines should also be seen in terms of the objectives they are hoped to achieve. Transductions describe how a particular activity is performed and results in a certain type of end state, and routines should be equally understood along these two dimensions, i.e. through their performance and through the reasons why they are being performed.

Technicity, or the role of technology in transductions, similarly links to the relation between applications and routines. Especially from the perspective of application developers, the idea of almost any application is to introduce technicity into the performance of a particular transduction or routine. Applications and routines interact in a particular location or situation, and together form a relationship that exists in a particular setting. What is important to see is that the two can be closely connected, where just as the routine is linked to the application, the application may have an impact on the routine as well. Dodge and Kitchin (2005) discuss this by using the concepts of code and space, and how the two can be understood by using the concept of transduction and technicity. They distinguish between different types of relations between code and space on the basis of the amount of technicity a particular transduction possesses.

For Dodge and Kitchin (2005), the importance of code in relation to space is decided on the basis of how essential the functioning of a code is for a particular transduction, and they find that transductions can be categorized as belonging either to a category called as *code/space* or then *coded spaces*. A transduction, where the intended transduction cannot occur without the code performing its part as intended is seen as belonging to a category of code/space. In code/space, code and space are seen as dyadic, which means that the relation between the two is mutually constituted up to a level that if one of them seizes to function, the intended transduction will not go forward. Examples of these kinds of events are train travel or watching movies online, both of which are dependent on the software to function. If the code does not work, neither of them can be done, or in other words the technicity of the code does not have the wanted impact, the relational problems cannot be solved and the pursued space cannot be brought into being. As a result, the occurring transduction leads to other types of end results and creates a change

into the overall chain of transductions, which would not have occurred if the code had functioned.

Coded space on the other hand is similarly a transduction where code has a role in mediating it but differs from code/space in the sense that code and space do not form a dyad. In practice this means that although code matters to the transduction of a particular space, the pursued space can still be largely brought into being even in the case the code does not function as intended. The non-functioning of code may impose certain costs or complicate the intended transduction to occur, but it is not the only solution for the relational problem in question. Code in coded spaces can be seen more as having a facilitating or augmenting impact, but not so much one of controlling and regulating. In other words, even though code has a role in the transduction, by not functioning it does not create an insurmountable obstacle for the desired transduction to occur (Dodge & Kitchin, 2005).

In addition to the two types of transductions, where code has a role in making the intended space to come into being, Dodge and Kitchin (2005) also discuss the so-called 'background coded spaces'. Background coded spaces can be seen as a sort of pre-existing state for code/space and coded spaces. The notion of background means that code can provide a solution to a relational problem but it has to be activated first. Examples of this can be found among coded objects and infrastructures like electricity but also radio signals or mobile phone signals that are always present, yet without being activated remain mute. When background coded space gets activated in a particular transduction, the transduction can either become one of code/space or coded space, depending on the impact the technicity of code has in that transduction. Overall, as more and more objects and infrastructures are becoming increasingly reliant on code and its functioning at least in some parts of their functioning, the scenarios of code/space are becoming also more likely.

3.4 From Societal Routines and Technological Resources to Code and Space

On the basis of the conceptual framework, applications can be described as intentions to introduce technicity to the routines or transductions that are prevalent in a particular location. Code and space function as the meeting point for the other key concepts within this research, namely societal routines and technological affordances. As a result, the conceptual framework discussed in this chapter points out the key areas where to focus on in explaining the role of context in application development. We propose the need to study the routines that the applications aim to address in their functioning, as well as to pay attention to the technological affordances given by the technological resources used in developing the applications. Routines

change due to changes in the surrounding context, but also the intention of the applications is to create changes to the existing context and to the routines stemming from that context. This change is mediated by the notion of transductions, which is where the technology gets intertwined with the existing routines in the form of the applications and their usage. Figure 3.3 describes this process, and the relations that exist between the key theoretical concepts that are discussed in this chapter.

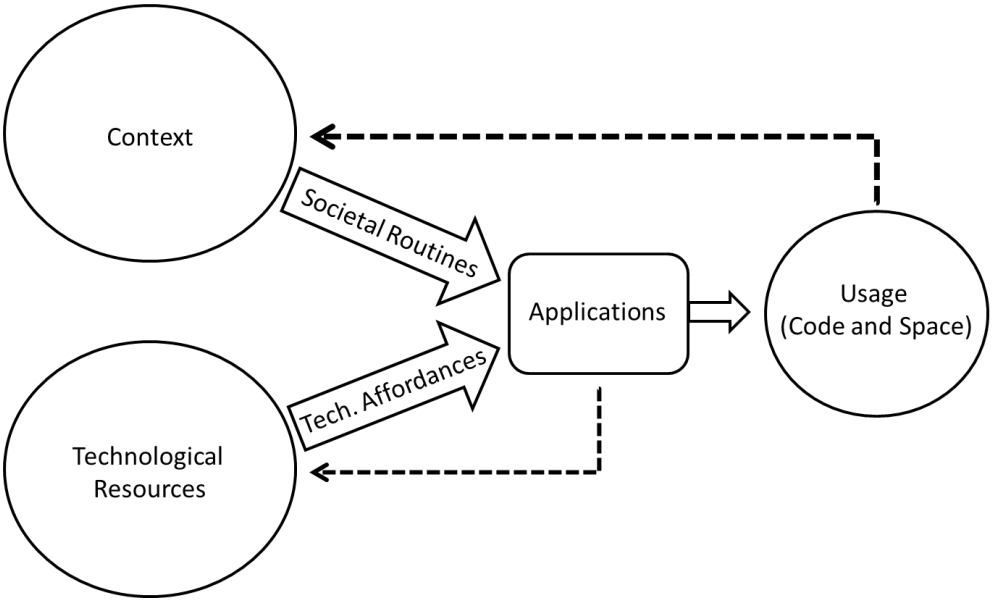

Figure 3.3. The Relation between Context, Technological Resources and Usage of the Applications

As noted above, the aim for this research is to understand *"how context impacts the application development process in a developing country?"* Each of the theoretical sections of this chapter highlights a particular aspect of that aim. Regarding societal routines, as those can be seen as proxies to the overall context, the question that arises is how and in which stages these routines can be seen in the process, and what is their overall role in it. As a result, the firs sub-question for this research and for the analysis to answer is the following:

Q1. How do societal routines display themselves in the application development process?

However, an application in the end is a technological construction, no matter the context. Therefore the role of technology has to be taken into account as well, and especially in relation to the societal routines that can be seen as impacting the application development process. As noted above, the developers have a role not only in building the applications but also choosing the technological resources that enable them to build the applications they want. The second sub-question therefore concentrates to the role of technological resources in building the applications, especially in terms of choosing them:

Q2. How are technological resources chosen to develop the applications and how these resources match with the developers' ideas for the applications?

Finally, for any application to have any kind of impact it has to be adopted and used by the targeted users. This stresses the need to understand the outcomes of launching the applications and the reception they receive from the users. This is also where applications are hoped to become integrated into the daily lives of their users and forming part of the transductions that these users encounter and are involved in. Therefore, the third sub-question aims to focus on what happens after the applications are launched:

Q3. How are the applications received by the targeted users and what kinds of implications this has for the applications?

In order to answer these three questions data has been collected and analysed for the research. In the following chapter, the methodological decisions made for this research will be discussed and the overall research design introduced. Also the location where data has been collected will be presented and justifications given why this particular research site is appropriate to answer the research questions set for this research.

4 Epistemological and Methodological Foundations of the Research

> "*An ontology without a methodology is deaf and dumb; a methodology without an ontology is blind. Only if the two go hand in hand can we avoid a discipline in which the deaf and the blind lead in different directions, both of which end in cul de sacs*" (Archer, 1995, 28).

> "*Scientific paradigm adopted by natural sciences is appropriate to information systems only insofar as it is appropriate for the social sciences*" (Hirschheim, 1985, 10).

4.1 The Ontological and Epistemological Basis of This Research

Developing an application is a socio-technical process, and the ontological and epistemological foundations must therefore accommodate both the so called material and non-material aspects of this research. What this means in practice is that these foundations need to understand the social processes that are involved in application development. As the research questions of this research show, the role of context is studied in relation to application development that occurs in developing countries. It has been argued by other scholars that in order to understand the context and the social processes involved in appropriating ICT, interpretivist methods are often deployed, and for example purely positivistic approaches may fail in grasping these factors in their complexity (Brown & Grant, 2010; Njihia & Merali, 2013; Walsham, Robey, & Sahay, 2007). In a similar manner, in relation to ICT4D field, it is important to understand the implications of the development context for the phenomenon that is being researched on (Madon, 1994; Prakash & De', 2007). However, at the same time they must be able to meet the requirements for understanding the technology, and not to let it "*fade into the background*" (Leonardi & Barley, 2010, 32), as Leonardi and Barley see that has been happening to some extent in information systems research. Not taking technology into consideration and seeing it merely as some sort of by-product of the social does not enable us to understand how technology is born in the first place and also how it may affect the social as technology conditions the responses that are available to its users (Kallinikos, 2010).

Traditionally, the interpretivist approach in ICT4D studies is linked together with social constructivism. According to Crotty, constructivism *"is the view that all knowledge, and therefore all meaningful reality as such, is contingent upon human practices, being constructed in and out of interaction between human beings and their world, and developed and transmitted within an essentially social context"* (Crotty, 2003, 42). Constructivism does not necessarily mean subjectivity in the sense that everything, such as the physical characteristics and forms of objects, could be constructed by people without paying attention for example to the physical laws that dictate the functionalities of these objects. As a result, there is a reality which in its

functioning retains certain autonomy from our thinking[10]. Regarding purely constructivist views, there is a risk that everything becomes social and socially constructed. Hacking (1999) stresses this point by arguing that social constructivism becomes as a term easily overused. While not denying that social constructivism takes place in different areas, in many cases what is being constructed is based on something that goes beyond the mere notion of social construction. Hacking gives an example in the form of a term on refugee woman and argues that even though the idea can be socially constructed, the terms also hold within factors that cannot be seen as merely constructed, i.e. real events that led the woman refugee to flee from her home country. Searle (1995) makes a similar argument with his distinction of institutional facts and brute facts, where he points out that there is a physical world that is not dependent on our existence. However, when facts or items require a human institution then they become institutional facts.

Constructivism here refers to the meanings that are given to events and objects, and is more about stating that the names, uses and understandings people give to different objects are products of the human mind and therefore constructed. Social constructivism on the other hand stresses the point of the social nature of constructivism, stating that these meanings are not decided in isolation by individuals but more constructed and negotiated in contact with other humans, and therefore, they are constructed socially and have a historical character. Regarding technology, one clearly social constructivist theory has been the social construction of technology (SCOT), which sees innovation as consisting of two sequential processes, interpretative flexibility and closure. The former concentrates in the groups that have a role to play in the innovation process, and who have differing interpretations on how the technological artefact needs to be built in terms of what aspect to prioritize. Closure on the other hand refers to the phase where the artefact is seen as ready, or alternatively gives rise to a new problem area (Pinch & Bijker, 1984). In terms of digital objects, closure as seen by Binch and Pijker is somewhat problematic, especially within the domain of this study. As Ekbia (2009) notes digital objects are unstable and unbounded, and have the tendency to resist reification. Nevertheless, even they can be argued of having moments of closure, for example in the event when an application is published.

Overall, this type of social constructivism is in line with an interpretivist analytical approach, which rose as a counterargument for positivist methodologies and focuses on the interpretations people make of social life. These interpretations have their basis in the surrounding cultural factors and are historically situated, which means that the social heavily impacts how people interpret and view certain social events (Crotty, 2003). Interpretivist

[10] As an example, if one drives a car towards a concrete wall with sufficient speed, there is going to be damage for the car no matter the interpretations given to the car and the wall

approach does not necessarily mean excluding causal links between events, it is more that what is seen as logical and sensible is largely context dependent, and therefore it is vital to try to view the world from the perspective of the people whose actions the researcher tries to understand, and to see the meanings that people give to certain actions, events and objects and the ways they interpret those (King, Keohane, & Verba, 1994). In information systems human agency often occupies a central role with its beliefs and motives, which has resulted in scholars arguing for more interpretivist methods and analysis over the more positivist approaches, as the latter held a strong foothold within the field (Smith, 2006; Walsham, 1995b). The argument for interpretivist research has been that instead of testing hypothesis according to the natural scientific paradigm, the research in information systems should place more attention to the contextual factors and the perceptions of the relevant actors (Avgerou & Walsham, 2000).

However, there is a risk especially on studies about technology that the research concentrates too heavily on interpretations and loses sight of the real qualities of the technological objects that also play an important role in explaining the causes of things. One approach to have both technology and the social taken into account in research is to merge them together in a manner that makes it difficult to tell where the limits between the two are. To some extent this is what has been done regarding the concept of sociomateriality, which stems from the ontology of agential realism. Sociomateriality focuses on practices, and argues that in terms of practices that involve technological and social components, the two can be seen intertwined up to a level that it is difficult to distinguish between them, and instead of doing so the two domains should be taken as a whole (Orlikowski, 2007; Orlikowski & Scott, 2008; Scott & Orlikowski, 2012). In other words, Orlikowski (2007) sees that *"the social and the material are to be considered to be inextricable related – there is no social that is not also material, and no material that is not also social"* (p. 1437). The positive aspect of this is the inclusion of technology into the studies, as it has been often left out for example in studies dealing with information technology usage in organizations (Leonardi, 2013; Orlikowski & Iacono, 2001). Drawing from Barad (2003) and Actor-Network Theory (Latour, 2005), sociomateriality argues that separating the social and the material and treating them as fundamentally different does not contribute to a better understanding of technology or its relation with the social, but that this separation has been artificially put in place by researchers themselves (Orlikowski & Scott, 2008). To some extent, this resonates with Williams and Edge notion on technology implementation in organizations. They state that "w*hen we begin to examine the implementation of technologies within organisations, we find that 'technology' and 'organisation' cannot be treated as entirely separate categories.[…] the definition of technology itself must incorporate the social arrangements within which it emerges and becomes embedded*" (Williams & Edge, 1996, 875).

However, these kinds of views are not without problems. Faulkner and Runde (2013) state that by not making a distinction between the material and nonmaterial objects, there is a risk of losing sight of the special characteristics that for example digital objects have. Leonardi (2013) and Mutch (2013) have argued that from an ontological perspective sociomateriality's founding on agential realism might be problematic, and mention four areas in sociomateriality where some other ontological and theoretical base might prove more suitable. First, it is not entirely clear whether some other theory such as ANT or socio-technical systems theory could not offer the same results as the usage of sociomateriality and agential realism. In addition from a methodological point of view, the issue of not treating the material and social separately might be problematic and difficult to do in practice. Third, agential realism also seems to ignore time by not taking into account how practices emerge and change over time. Last, borrowing from Faulkner and Runde (2012), sociomateriality does not seem to allow the existence of practices where the relation between the social and the material are external, i.e. the existence of the two does not automatically contribute to the creation of a new practice. As an example, the use of a computational tool does not make a weather scientist, but neither does the weather scientist make the computational tool.

As Leonardi and Barley (2010) note, it is important to see how technology and social affect each other as distinct categories. Technology allows users to do certain things while disabling others, and social process shape the development of technologies and their effects. Overall, none of this is to deny the interconnectedness between the social and the material as such. Leonardi (2011) for example refers to the relationship between the social and material as one of imbrication, which is seen as describing the interplay that exists between the social and material that eventually leads into a durable infrastructure. However, in imbrication the two domains are dealt with as separate: for example the social agency can be discovered in the social group's objectives, and the material in terms of the material performing in ways not entirely controlled by the group working with the material. Over time these two factors become entangled in the infrastructure through a process of overlapping and interlocking elements, yet differ in their relation to intention.

Even though it can be claimed that the social world and the meanings ascribed to technologies are largely socially constructed, technology and technological resources still have a basis in real natural laws that can be more difficult to alter, apart from finding out in more detail how they work and how they are connected to other events. Borgmann (2010) sees technology as a transformation of reality, which is based on ontology that in addition to being among other things historic and dynamic, is also enclitic. By this he means that technology "*is an enclitic ontology because technology does not reconstitute reality entirely. The device paradigm has to*

lean on resources not of its making. Resources in the first instance are raw materials and energy. More revealing, traditional culture is a resource as well. It has lost its former moral force, but not its characteristic texture that comes to be deeply imprinted by technology" (p. 31).

To capture both the characteristics of technology and the implications of context, this research has to find a way to accommodate the social and the technological in an ontologically and epistemologically sound manner. Interpretivist approaches by large seem to reject the notion of causality, or view it as multidimensional or circular, which to some extent then leads to questions on the usefulness of interpretivist research in the first place if all it can provide is context-specific descriptions without generalisability (Smith, 2006). Walsham (1995a) sees that generalizations within interpretivist research are possible, but they should be seen more as tendencies than predictions. However, some causality might be seen for example in the role that technological resources play in application development: not everything can be built, and what can be built are to some extent dictated by the resources. This also further highlights the need to incorporate the notion of real into the research. Sismondo (1993) however argues that in terms of ontology and epistemology, the social constructivist view and the realist notion of the real are not incompatible, as long as one is ready to place limits on how much of the world is socially constructed leaving space to the notion of real. Crotty (2003) also points to this direction by noting that "*constructionism in epistemology is perfectly compatible with realism in ontology*" (p. 63), at least mild versions of constructionism, which as mentioned accepts the existence of real instead of seeing everything socially constructed.

Critical realism is well suited for adopting this kind of stance as it includes parts of the constructivist approach while maintaining notions of real in place. Smith (2006) makes this connection explicit by stating that critical realism "*allows for the pursuit of an interpretivist agenda without denying the existence of the subject under study or its role in regulating research*". Mutch (2013) sees that critical realism enables the researcher to take into account the question of time, and view the research object as an ongoing process or construct that occurs when material and social interact with each other. This way the material and social also keep their distinctiveness, where the material displays itself as a realm of structure that is in interaction with the realm of action, which refers to the social and its role in the equation (Leonardi, 2011). Furthermore, it leaves room for causal explanations without arguing that certain events would always trigger similar outcomes, which is something that can be seen in the four key principles critical realism captures (Sayer, 2000). First is the notion made by Bhaskar (2008) on the intransitive and transitive dimensions of knowledge. Sayer describes the difference by marking the distinction between theories and the world they are about. Theories are seen as transitive and subjects to change, whereas the world they are about is considered

intransitive by the critical realists, in other words the world does not change with the theories. This resembles Searle's notion on institutional and brute facts. The second principle refers to the concepts of real, actual and empirical. Klein (2004) summarizes the three by stating "*the real are the causal mechanisms and structures that produce actual events a subset of which then is empirically observed*" (p. 131). The argument goes that the real cannot be seen, but consists of causal mechanisms that impact actual events. A certain part of these events can be observed, and on the basis of what can be observed one is able to make inferences on those underlying mechanisms and how they work. The third principle is related to the critical realist notion of emergence. It refers to a situation where two or more factors conjunct creating new phenomena, which has properties that cannot be reduced to factors which were needed to create it, and in a sense this also implies that the phenomena itself is more than just the sum of its components. The last principle mentioned by Sayer is the one of causation. The concept of causation is not understood merely as effect following cause. Causes are dependent on multiple factors, and therefore same effect may not lead to a same cause. Causal mechanisms are affected by the structures and powers that the object itself is tied. These powers can be activated or they may remain passive.

What this means is that there is a reality irrespective of human beings and their perceptions, but it cannot be observed directly but displays itself through structures and mechanisms in the form of real life events and other observable factors. Therefore there exist a particular notion of causation, but it is less straightforward than the one supposed by positivists, as mechanisms may also be linked to powers that in one particular situation remain inactive but in another do the opposite, changing the outcome. The real sets the scene on what can be done and also places limits on this, yet does not mean that only particular mechanisms would lead to specific outcomes, leaving therefore room also for the social construction of events and artefacts. Overall, Crotty (2003) defines epistemology as "*the theory of knowledge embedded in the theoretical perspective and thereby in the methodology*" (p. 3). Above presented notions on the connection between technology and the social can be transferred and studied from the perspective of constructivism while keeping critical realism as the underlying ontology for the research, and thus allowing the existence of the real.

Overall, constructivism in some form or another seems a common approach in ICT for development research. Walsham et al. (2007) mention that interpretivist research using qualitative methods are often seen as the preferred way to go forward with research focusing on ICT and ICT implementation in developing countries. Critical realism does however suppose an existence of real in a manner that not everything can be interpreted. However, it also leaves room for constructivism and to some extent the notion of interpretivist approach but argues

that there must be a way to settle which interpretation of the events is more accurate (Easton, 2010). Overall, within the confines of symbolic interactionism, the relationship between interpreting and the real has been described through four notions developed from Mead's writings. According to Blumer (1980), the first one states that "*there is a world of reality 'out there' that stands over against human beings and that is capable of resisting actions toward it*" (p. 410). Second, "*This world of reality becomes known to human beings only in the form in which it is perceived by human beings*", and therefore "*this reality changes as human beings develop new perceptions of it*" (p. 410). Finally, "*the resistance of the world to perceptions of it is the test of the validity of the perceptions*" (p. 410). This combines well with the critical realist notion of reality, yet allows also investigating the social processes that are included in the development of mobile applications. It makes also possible to discuss the material and non-material factors at the same time without denying the existence of either of them.

Combined with constructivism as epistemology, critical realism provides a suitable ontological framework for this research, where the central actors are the application developers, who function as the unifiers between the surrounding contextual factors and the technological affordances provided by the different technological and other resources. Lévi-Strauss discusses of bricoleurs as persons who are able to make something new out of materials that had been previously formed a part of something else. A bricoleur takes a particular material, and turns it into something new (Lévi-Strauss, 1966). Because of the digital nature of the tools and resources used in application development, it cannot be really claimed that the developer is taking something existing and turning it into something new. However, the logic largely applies to application development as well. What the developer ends up making is enabled and constrained by the physical and material limitations of the technological resources. The material is often subject to physical laws which cannot be so easily altered, or as Mutch (2009) argues, *"although hardware is socially shaped and bears the marks of assumptions about the nature of the world, these are embedded to a degree that is more resistant to change and appears more 'natural' than aspects of the software and data structure*" (p. 511). The logic can be also described through Feenberg's (1999) notion on primary and secondary instrumentalization: the primary instrumentalization abstracts dimensions of reality and passes them on to the secondary one, where the technology is more subject to design choices and shaped by the social processes. As a result, the developer can interpret the material and construct whatever she or he wants, but always within the limits of the real. Within those limits, the developer however has considerable room in constructing what she or he wants.

Critical realism has been used also elsewhere in information systems literature and also in ICT4D research. As noted above, critical realism provides a sort of middle ground between the

traditional positivist and interpretivist camps (Smith, 2006). In a critical realist study, the aim is to distinguish generative mechanisms, which under certain conditions may trigger the type of events that are then later on witnessed by the researcher on the basis of the empirical. Although there exists causality in the ways mechanisms lead to the emergence of events, this causality is not positivist in the sense that same instances do not always trigger the same mechanism. Discovery of generative mechanisms is seen as especially useful in relation to the creation of so-called middle range theories, which escape some of the abstractness of grand theories and are better positioned for the creation of new and novel insights regarding the topic areas in question (McGrath, 2013, 2016). For this research, the critical realist notion of generative mechanism provides an analytical tool, which allows to explain in more detail how context impacts application development. At the same time, it enables the research to focus on the key areas that are crucial in understanding this connection between the two concepts and to theorize further on how context and application development are interlinked.

Overall, the strong point of critical realism is that it allows the notion of real and causal mechanisms without slipping into the realist view that takes the world as we see it and. At the same time, critical realism leaves space for constructivist understanding on how the social affects technological artefacts not only in terms of usage but also in their creation (Smith, 2006). Therefore this research adopts critical realist ontology and combines it with a so called mild constructivist epistemology, which does not deny the existence of real or reality that exists irrespective of us.

4.2 Case Study Research

This research was performed as a case study on the application development process in a particular area, namely in Kampala, the capital of Uganda. Case studies can either assume interpretivist approach or take a realist view of the world (Yin, 2014), and have been used to study information systems development and use especially in organizational contexts (Darke, Shanks, & Broadbent, 1998). As discussed above, although a particular society is hardly a traditional type of organization, within the confines of this research there are similarities due to the usage of the concept of routines, which has been borrowed from organization science literature. Furthermore, as the literature review pointed out, application development in developing countries can be seen as a relatively new phenomenon, and case studies have been seen as a good method to do research on areas that have not been discussed to a great extent in the literature (Benbasat, Goldstein, & Mead, 1987). Overall, according to Yin (2014) case studies are also suitable for studying topics that cannot be easily distinguished from their context, which is very much the case regarding this research. Case studies have also been seen as useful in terms of theory extension and development (Flyvbjerg, 2006), which is one of the

objectives set for this research. Due to these reasons the case study was seen as a suitable method to answer the research questions set for this research.

In general, case studies are used to answer research questions that start with the word 'how' or 'why' (Yin, 2014). While there are many ways of defining 'case study', Yin identifies two different perspectives, one referring to the scope of the case study and the other to its features. According to him, *"a case study is an empirical inquiry that investigates a contemporary phenomenon (the 'case') in depth and within its real-world context, especially when the boundaries between phenomenon and context may not be clearly evident"* (p. 16). Regarding the features of a case study, Yin sees a case study something that deals with situations where there are many variables of interest and not just data points, relies on multiple sources of evidence that enable triangulation and also should take advantage of a theoretical proposition, which will guide the researcher through the phases of data collection and analysis.

The contemporary phenomenon in this research is application development that occurs in a particular area in a developing country. Similar events have been seen in other developing areas as well, and in Sub-Saharan Africa Nairobi is often seen an example of an area that aims to develop a thriving application development and software industry (Gathigi & Waititu, 2012; Graham & Mann, 2013; Vision 2030, 2007). However, at the same time it is not clear what is the relationship between contextual factors and application development or where the boundaries lie between the two, which is also an area this research aims to contribute in. As could be seen from the conceptual framework, routines, which in this research provide a lens to understand context, are not straightforward and can be rather complex to describe, and as a result cannot be explained in terms of few data points or isolated variables. This also matches with the features for case studies as described by Yin, and was also one of the reasons why different sources of evidence were used for the research. Overall, the theoretical position provided in the conceptual framework supported and guided the data collection and analysis.

According to Yin (2014), there are different types of case studies that a researcher can employ: single case study, single case study with embedded units of analysis, multiple case study and multiple case study with embedded units of analysis in each case. This research adopts the form of a single case study, but to some extent it uses the approach of embedded units of analysis. The case overall is the application development scene in Kampala, but the central units for this research are the developers and start-ups that reside in the area, which in certain way are embedded into the case. The single case study method was chosen for two reasons, due to the commonness or typicality of the case and also because of its longitudinal aspect (Yin, 2014). Firstly, it was seen that the chosen case provided the researcher a common case of a particular

phenomenon, namely developing applications in a developing country context. Second, the data collection tried to also follow up what happened to the units of analysis over the time period of one year.

A single case study with embedded units of analysis has its challenges, the main one being that the researcher does not return back to the main level of analysis once having focused on the sub-levels. In addition to this, and especially regarding single case studies, generalizability is often an issue (Yin, 2014). Overall, according to Walsham (1995a) single case studies are useful in relation to four types of generalizations: development of concepts, generating theory, drawing on specific implications and the contribution of rich insight. This research aims to provide a particular way of understanding context and by doing so contribute to theory development, and it is seen that therefore generalization, although in certain ways problematic, should not be seen as a major obstacle for performing a single case study to meet the objectives set for the research. Yin (2014) sees that case study overall works well to use theories as templates against which to compare the results obtained from the research, and this type of analytic generalizations can enable the researcher to form theories that have relevance also outside the single case chosen for the research. According to Tsang (2014), this view can be further strengthened by using a critical realist ontology, which aims for the identification of particular mechanisms.

In relation to critical realism, a case study aims to understand the mechanisms that generate the outcomes, or the empirical, that are accessible and visible for the researcher. This research aims to display the mechanisms that can be seen as reasons for the outcomes that emerge from the data. This process has been labelled as retroduction, and to some extent also shares similarities to interpretivist approach that uses explanations and observations of events and actions as the tool to conduct their analysis (Mingers, 2004). Wynn et al. (2012) note that within critical realism *"causal explanation for a given phenomenon is inferred by explicitly identifying the means by which structural entities and contextual conditions interact to generate a given set of events"* (p. 787), and that overall the ontology is suitable to be used in case studies to study the interaction that occurs between actions, events, context and structure and explain them through causal mechanisms. This resonates well with the objectives of this study, which aims to understand the role of context in application development.

Wynn and Williams (2012) also give further recommendations on how to carry out case studies that have adopted a critical realist ontology. Their methodological principles for critical realist study consist of five points, where the first is explication of events. With this they refer to the need to *"identify and abstract the events being studied, usually from experiences, as a*

foundation for understanding what really happened in the underlying phenomena" (p. 796). The second principle, explication of structure and context, aims to identify the key components of the contextual environment, social and physical structures, and the relationships that exist among them. The principle of retroduction aims to distinguish the powers within the structures and context that have enabled to generate the events. Empirical corroboration, which is the fourth principle, establishes the requirement for the research to *"ensure that proposed mechanisms have causal power and that they have better explanatory power than alternatives"* (p. 796). The last principle, triangulation and multimethods, stresses the need to use several data sources and analytical methods to support the causal analysis.

These principles should not be seen as isolated or to be followed in a sequential order, but are dependent and need to be checked and kept in mind throughout the research project, especially during the data collection and analysis stages. Overall, they provide this research a methodological framework for drawing inferences from the collected data that will enable to answer the set research questions. The data collection was guided by the conceptual framework developed for this research. The following section will explain this in more detail.

4.3 Conceptual Framework and Data Collection

The conceptual framework for this research lays out a model where technological resources and societal routines are being coupled together in the form of an application, which is then offered to the users. If the users adopt the applications and use it, the application can be seen as becoming part of the coded spaces that are present in the location where the applications' targeted users reside.

In order to answer to the research questions set for this research, it was therefore necessary to try to see the points where either the technological resources or societal routines surfaced during the lifelines of the applications. The intention therefore for the data collection was to try to map these lifelines and by analysing the data to see where in the application development process one of these two factors played a role and how. Related to the notion of code and space, the aim was also to go a little bit further by getting insights from the targeted users in order to have data on the perceptions and opinions the users held on the applications developed locally. However also the developers and start-ups themselves functioned as a certain types of proxies to the users, as the role of users and their views of the applications were closely linked back to the overall development process of the applications.

In order to map the lifelines of the applications, the development process of an application was divided into different stages. The first one was about the ideation stage of the application, and the intention here was to study where the ideas for the applications had come from and how

they were shaped in the process. The second stage concentrated more on the actual development of the applications, where the initial versions as well as later updates were then built into the applications. The third part consisted of looking at the launch of the applications and the reception they had received among the users, and built on top of the previous stages, as it tried to further understand what kinds of strategies were developed in case the application had not been able to gain popularity among the targeted users. Needless to say, these stages are seldom this separate, and in many cases it even might be that several of the stages are occurring at the same time. However, the findings are presented in this manner for the sake of clarity.

Overall, the data analysis, which shall be discussed in more detail later in this chapter, consisted of looking from these different stages of application lifelines references that were made regarding existing societal routines or to technology overall in the form of devices, digital resources like SDKs and APIs, or skills. This enabled the author to link the data collection and findings back to the research question on the role of context in application development. The intention for the data collection was therefore to find the context specific mechanisms that affected application development in a particular geographical area.

What is noteworthy is that in order to do this, the research did not concentrate on particular type of applications, but tried to capture a wide array of different types of applications. The purpose for this was to have a more holistic understanding of the particular contextual mechanisms that were common for a large variety of start-ups and applications, and not only for example health related applications. As a result, in the findings and analysis phase, the experiences of the different developers and start-ups were grouped together to find the common aspects that these case study units of analysis encountered. In addition, during the data collection phase collected data was used to compare how far similar experiences were encountered across the different units.

In order to do this, different kinds of data collection methods were used, which will be further presented in the following section. Among other things, various data sources enabled the author to seek verification for the found references for technologies and societal routines, and have a clearer view that the references were also at least relatively common across the developers and not only isolated to one or two developers or start-ups.

4.4 Data Collection

For carrying out case studies, Yin (2014) discusses certain principles that have to be met. First principle states the need for using multiple sources of evidence. As shall be seen below, this research used a combination of interviews, non-participant observation and document

gathering as methods for data collection. Although the main bulk of data consists of interviews, the other sources of data were used for triangulating the findings stemming from the interview data. The second principle, creating a case study database, consists of organising the data in the form of a database, which can be further used to write the case report. Whereas the database can be seen as a collection of the data collected, the report takes a step further by including the interpretations of the researcher that has been done on the basis of the data. Yin's (2014) third principle, maintaining a chain of evidence, is a tool that enables to increase the reliability of the research and enables the reader to follow the path that the researcher has taken from formulating the research questions to the conclusions of the research. Overall it consists of several steps, and states that the conclusions of the analysis have to be able to also show how they are based on the findings. These again are formed by using the data collected for the research, and this data is collected according the case study protocol, which basically states what kind of questions were asked from the interviewees, what were the aspects that were paid attention during the non-participant observation and what type of documents were sought after and why. Finally, that protocol is based on the research questions set for the study. As noted above, the purpose of this chain is to ensure the reader is able to follow where the conclusions of the study originate from and how they were reached. The final principle, being cautious when using data from electronic resources, is less of a concern within this research, as only the documents come from electronic resources and even those are taken directly from material produced by the developers and start-ups themselves.

Conducting a case study that relies on multiple sources of evidence increases internal validity of the research (Benbasat et al., 1987; Yin, 2014). In this research, three types of data were collected, interviews, observations and documents using qualitative methods. Overall the usage of qualitative methods requires a certain amount of subjectivity in the research process, but it has been also noted that this type of subjectivity should be seen as a source of strength rather than as a weakness. This is because no research is value-free and this type of recognised subjectivity enables the researcher to have a deeper understanding of the relevant actors' perspectives and views (Garcia & Quek, 1997).

The material for the research was collected mainly on three occasions over a total period of four months that were spent in Kampala, Uganda. The first visit took place in July 2015, the second one from November to December the same year and the last one from June to August 2016. Most of the time during these trips was spent in two technology hubs in the area and overall by participating in different types of relevant events that occurred in the area during that time. In addition to this, Skype interviews were made with the developers outside these dates to gather

more data and also to ask more specific questions in case there were areas where greater clarity was needed.

In relation to data collection, there are two points regarding the relation between the research location and the researcher's own background that are important to mention. First of those relates to the positioning of the researcher, as in some cases there was a tendency by the interviewees to see the researcher as a possible investor, which had the possible effect of impacting the answers given to the researcher. In order to avoid this, the researcher explained in detail before the interviews what the interviews are for and in some cases also stressing the point of him being interested in the start-ups only because of his research and not as an investor. In addition, the intention was to create a friendly and informal environment for the interviews, which was hoped to further clarify any doubts regarding the researcher's role. Related to this and especially to the observation part of data collection, being often the only white person in the tech hubs, the researcher's ethnicity might have somehow impacted the overall daily functioning of the hubs. There was no indication of this as such, apart from the possibility of seeing the researcher as a possible investor, regarding which the above explained measures were taken to avoid this from occurring.

The second point is related to the fact that the researcher was not very familiar with the location in the beginning of the fieldwork. Although it complicated things for example by not being aware of the local customs, this however was not an entirely negative factor either. The non-familiarity enabled him to pay attention to areas and things that otherwise might have gone unnoticed for someone more accustomed to the location. This type of effect has for example been present in the field of anthropology (e.g. Roberts, 1989), where unfamiliarity not only makes the researcher unaware of the functioning of a particular context but in some ways can also help the researcher to notice important factors and occurrences that someone more local would easily take for granted without giving those too much thought.

4.4.1 Semi-Structured Interviews

Walsham (1995a) notes that in interpretative case studies, where the researcher portrays her- or himself as an outside observer, interviews often function as the primary sources of data. Interviews enable the researcher to access participants' interpretations of the events and actions that are of interest as well to capture their views on the relevant matters. This is also the case in this research, where interviews formed the main bulk of the data collected.

Interviews are seen as a relationship between the interviewee and interviewer. In this sense they are more like conversations, where both participants are involved. They can also be seen as contextual and relational, where also the interviewer is involved in the process of meaning

construction (Kvale & Brinkmann, 2008). Within this paper the interviews were not taken as objective descriptions of reality, but the interviews were seen as situations where an interviewee gave his or her view of the relevant events and actions (Marshall & Rossman, 2006). This however does not mean that the interviewees were seen as not telling the truth, but more that the subjectivity was acknowledged in the interviews. This also fits with the critical realist ontology that is combined with interpretivist or mild constructivist epistemology, as the events and actions, although based in the real, are subject to interpretations.

This however demands effort from the part of the researcher, and Myers and Newman (2007) mention certain guidelines that need to be followed. For example, the interviewer has to situate her- or himself as an actor in the interviews. Within this research, it helped the researcher to have some experience in software development. On the other hand, in most cases the different ethnicities between the interviewer and interviewee highlighted the need for the researcher to clearly explain his role as a researcher and not for example as a possible investor. Secondly, and to some extent related to the first point, the researcher should try to minimise social dissonance and try to make the interviewees feel as comfortable as possible. As Myers et al. (2007) note, this was done by trying to give good first impressions, making the interviews more conversation-like, and by trying to adopt expressions and language that the interviewees used. Furthermore, the interviewer tried to fulfil the requirement to *"represent various voices"* (p. 17).In addition to interviewing developers, we interviewed users of the applications, known technology bloggers in the area, directors of technology hubs where many of the start-ups resided as well as relevant institutional actors such as university lecturers and researchers that were involved in the application development scene in Kampala. The last point also helped with the *''triangulation of subjects''* (Rubin & Rubin, 2011), which leaves room for different opinions and views to rise.

On another note, although conversation like, the researcher in this research had prepared for the interviews by creating topic guides (Table 4.1) for them, yet leaving room for other topics to emerge as well. This type of semi-structured interview offers a middle ground between structured and unstructured interviews. They follow a certain sketch, but allow the interviewer to ask additional questions as well as for the interviewee to be freer to explain things that might not seem relevant at first sight. The interviewer must sometimes allow the interviewee to lead the discussion (Esterberg, 2002). Myers et al. (2007) note that semi-structured interviews together with unstructured ones are the most typical in information systems research. In a *"semi-structured interview there is an incomplete script. The researcher may have prepared some questions beforehand, but there is a need for improvisation"* (p. 4).

The topic guides presented in Table 4.1 provided a sketch for the interviews, but in case different topics emerged during the interviews, those were also followed. Naturally, the questions varied according to the interviewee, and also in case the interview in question was more of a follow up on how the start-up was doing, the questions focused more on the technological and other changes that had occurred since the last interview, as the intention was to capture all the latest developments regarding the application and the start-up itself.

Application related questions:
Could you explain in your own words what type of application you are developing/have developed and how does it works?
How did you get the idea for the application?
How did you choose the platform to build the application on? Why did you choose that particular platform?
What kind of other technological decisions were made regarding the application and what was the reasoning behind those?
How have the users adopted the application?
Have you collected any feedback from the users? If so, what kind of feedback have you received?
What are the near term future plans regarding the application for example in terms of adding functionalities or trying to reach new market areas?
Overall, how has the application development process been so far?
Personal background and general views:
What is your view of the application development scene in Kampala overall?
What are the major opportunities and obstacles?
Have things changed during the last years?
Which applications you consider as particularly promising and why?
Do you think many of these companies overall are viable?
How do you see the role of the technology hubs and other relevant actors like government or universities in helping start-ups and application development companies?
How did you became an application developer (or in case the interviewee was not an application developer, then that person's profession/role) and involved in this project?
What is your educational and professional background?
Besides building applications, do you do any additional work?
Users:
You know many Ugandan applications?
Do you use any Ugandan applications? Any other applications?
If you know, but don't use them, why not?
If you are using a local application, what is your opinion on it?
How did you start using it?

Table 4.1. General Topic Guides for the Interviews

In total 64 persons were interviewed for the research, out of which 35 were with the developers or other members of the start-ups, and some of those consisted of several interviews over time. On top of those, 11 hub directors, representatives of institutions like universities and aid organisations as well as other relevant actors like local technology bloggers were also interviewed. Finally, 18 interviews for targeted users were also carried out to find out their views of the applications. For each of the interviews consent was asked as well as the purpose of the

research explained. It was also made clear that at any point, if the interviewees so wished, they could decide to withdraw the information given in the interview.

In general, the intention with the questions was to map the contextual and technological dimensions of application development process and capture the views different actors had on the scene in Kampala and their understanding of the application development occurring in the location. These interviews were then backed up by observing the relevant actors in their work and gathering documents that were seen relevant for studying the application development scene in Kampala. This also served to fulfil the requirement of case studies of using multiple sources of evidence as a way to triangulate the data.

4.4.2 Non-Participant Observation

Observation as a data collection method can be divided into participant and non-participant observation. The idea behind non-participant observation is that it aims to observe events without actively participating in them and in this sense also refrain from affecting these events one is observing (Robson, 2011). However, it has been noted that sometimes the pure presence of the researcher changes people's behaviour. Esterberg (2002) also claims that already the fact of the researcher having experiences of a certain situation leads the researcher to interpret the situation in a particular way. Due to these reasons it has been said that every type of observation is participant observation, which to some extent also applies to this research as it is recognised that the mere presence of the researcher possibly had implications for the observations. However, the general view is that in order for a researcher to perform participant observation, she or he should be engaged in the activities that the targets of the observation perform (Robson, 2011). This was not really the case in this research, and therefore the observation was more of the type of non-participant observation. Overall, within the confines of this research, observation took place mainly in two technology hubs in Kampala, as well as to some extent in different kinds of meetings, workshops and pitching events which were organised rather regularly. Regarding the users, the researcher was also able to observe how users engaged with the applications. According to Marshall and Rossman (2006), the main point in observation is to find out the patterns and events that keep on occurring, but it is noteworthy that just like interviews, observations are also subjective as neutrality cannot be obtained due to the fact that the researcher chooses where to focus her or his attention (Clifford & Marcus, 1986). However, to some extent observation enables the researcher to see how meanings are made and how people react and interact in the face of certain events, although as pointed above, the researcher also becomes part of this creation of meanings. In this case observations enabled the researcher to witness how start-ups worked, how they interacted between themselves and in

the events where they went to look for further funding. However, it must be noted that in an open office environment there were often moments where there was literally not much to observe, as many of the developers spend their days programming or coding in front of the computer. This also had the effect that it was necessary for the researcher to be active in his observation, which in practice meant for example asking questions from the developers about what they were working on at a particular moment and engaging in conversations with them.

In addition to the developers, time was also spent with the targeted users of the applications that often took place in more informal settings outside the offices. Similar to the developers, questions were asked during the process of developing the applications and in many cases, observation was combined with informal interviewing. However, in some cases it was possible to observe how users of particular applications, such as motorcycle taxi drivers, used the application in their hangout posts. In addition to this, as the research focuses on context, also the time spent in Kampala was valuable in trying to get a better understanding on how the society worked and how people in general went on with their lives. This in no way is to make the claim that the researcher was able to fully understand the local society or comprehend all the aspects of it, although it did provide a better view how different events and actions were linked to the overall contextual canvas of the society.

Overall, the observations were written down in notes and on the basis of those notes, occurring themes were looked for and interpreted. Especially in cases where there was doubt about their meaning, questions were asked from the relevant actors in an attempt to clarify the aspects the researcher had noted and to see whether the researcher's interpretations matched with those of the subjects. This was not always straightforward, as the researcher avoided asking directly whether the relevant actors shared his understanding yet was eager to get their view of the events and to see how well that matched with the one of the researcher.

4.4.3 Document Data

As Esterberg (2002) states, a document refers to *"any written materials that people leave behind"* (p.121). Documents are not research-generated, but treated as found, and can in certain ways be seen as more concrete than the other data collection methods used in this research. However, Myers (2004) also notes that even documents are ambiguous and do not necessarily present any objective world, as they are being created by someone with particular objectives in mind and should be seen through hermeneutics. In this research, documents that were gathered were texts that were written by developers of the application providing details on what it was supposed to do. Examples of these types of documents were the update lists that were published on the application's download page providing information on the type of updates that

had been done, and instructions that were given to users on how to use the application. However, these types of documents were not available for every application, but where available they were used to back the findings from the interviews and non-participant observation for in relation to track how applications had changed, to see how applications had performed in terms of downloads and also to evaluate the type of trainings the technology hubs provided. They also functioned as material for questions that were asked in the cases where there was a possibility to talk to a developer or to another person involved in the start-up on multiple occasions over time.

4.5 Data Analysis

Miles and Huberman (1994) see the process of analysing qualitative data consisting of three different stages: data reduction, data displays and finally conclusion drawing and verification. This process occurs in largely sequential order, where the researcher is expected to move from one level to another and while doing so, also increase the amount of abstraction.

The first stage, data reduction, is about *"summarizing and packaging the data"* (Miles & Huberman, 1994, 92), in which the collected data is transformed into a form that it can be further analysed. In practice, this meant transcribing the interviews as well as transform (in case it was not already so) the collected field notes and documentation into written text. The data was then organised into units according to the application they were related to. If the data was more about general views that affected application development within the region, as was the case for example with interviews with technology bloggers or technology hub directors, they were organised into a group of their own. The data was then further divided into three stages (ideation, application development and application usage), which as a whole is seen as capturing the lifeline of an application.

In the second stage, data displays, the intention is in *"repackaging and aggregating the data"* (p. 92), which refers to the search of themes that emerge from the data (Miles & Huberman, 1994). In order to do this, thematic analysis was used, where the aim was to look for references regarding societal routines or technological factors affecting the application development. After identifying particular themes and data that functioned as evidence of these themes, these pieces of data were then re-organised once more according to the stages of application development that they came closest to. Within these stages, the data was further organised into sublevels according to the theme that these pieces of data presented or were an example of.

The third stage on conclusion drawing and verification aims for *"developing and testing propositions to construct an explanatory framework"* (Miles & Huberman, 1994, 92). The themes

that were discovered in the previous stage were then traced back to the research questions and used to answer them. The propositions were largely developed during the first two fieldtrips and the testing was mainly done in the third stage by asking from the interviewees their opinions on the developed propositions. This also lead the researcher to amend some of the propositions and correct them in some cases based on the views of the interviewees. Especially regarding the more macro-level propositions, discussions with hub directors or technology bloggers proved out helpful. These propositions were further used to create an explanatory framework that laid out the suggested mechanisms that are seen as answers to the set research questions. In identifying these mechanisms, Wynn et al.'s (2012) principles on how to identify mechanisms in critical realist case study were used to provide guidance for the process.

4.5.1 On Thematic Analysis

Thematic analysis enables new emergent patterns to grow out of the data, yet at the same time it allows the usage of external theories to guide the interpretation process (Attride-Stirling, 2001). Overall, thematic analysis seems to include some aspects of grounded theory approach without the requirement to only rely on the data in theory building. Flick (2009) notes that thematic analysis, especially in terms of coding, is an on-going process in the sense that the researcher may find new themes along the way and also might find it necessary to adjust the existing ones. In that sense, the emerging themes can be seen as alive and evolving throughout the whole process of analysis and data collection.

Using thematic analysis it is also possible to see how the underlying structures affecting the events are produced, or how the informants themselves interpret the events under research. It could be argued that discourse analysis can also be used to gain deeper insights into the topic and see how the data represents the informants' views of the world. However, discourse analysis is not really seen as suitable method for this research, since despite the intention of trying to see some of the underlying structures that can be found from the data, there is no aim to see how the informants and others use certain kind of expressions or how they construct reality through their usage of these expressions (Hammersley, 2003).

The researcher has to be aware on how he or she may affect the analysis. The themes that emerge from the data are first interpreted by the researcher, and therefore there is a risk that the researcher sees what either he or she wants to see, or is unconsciously guided by his or her cultural or social background. In other words, the key point in thematic analysis is to find the patterns that are in the data, and while doing so, the researcher should be aware of how she might end up affecting the process (Braun & Clarke, 2006). Thematic analysis that is based on a theoretical framework helps the researcher to overcome this type of interpretation bias to a

certain extent, since it is not possible to entirely ignore one's background and its implications for the analysis. Another aspect that also allows the researcher to overcome some of these obstacles is simply to be aware of the issue (Braun & Clarke, 2006).

4.5.2 Themes Found from the Data

As discussed above, the guiding principle through the analysis was to map the lifelines of the applications and to find the relevant themes that match with the different stages of application development. An additional level was also added to determine whether the themes were more related to the technological factors or to the routines that stem from the local context.

STAGE OF APPLICATION DEVELOPMENT	SUBAREA OF THE STAGE	THEMES EMERGING FROM THE SUBAREAS	THEME'S MAIN CONCEPTUAL BASIS (ROUTINE/TECHNOLOGY/ BOTH)
IDEATION	Ideas for the applications	Origins of ideas	Routine
		Access to developers	Technology
		Other factors affecting the ideas and content of the applications	Routine
		Funding and monetization	Both
	Refinement of ideas	Perception verification	Routine
		Misunderstandings regarding the idea	Both
		Technological constraints	Technology
BUILDING THE APPLICATION	Choosing the technological basis for the applications	Access to developers	Technology
		Process	Technology
		Knowhow and skills	Both
		Institutional background	Routine
	Building the applications	Location	Both
		Constraints in building the application	Routine
		Increased technological resources	Technology
		Language	Routine
LAUNCH AND USAGE OF THE APPLICATION	Launch of the applications	Delaying	Technology
		Usage	Routine
		Marketing	Both
	User feedback	Relation to users	Routine
		Incorporating feedback	Both
		Scaling up and updates	Both
	Adaptations and updates to the applications and to the business models	Change of technological basis	Technology
		Targeting new user groups	Routine
		Application as a side product	Routine

Table 4.2. Themes Emerged from the Development Stages and Their Conceptual Basis

Table 4.2 provides the coding scheme of the themes that were found, as well as lists what was seen as their conceptual basis and how they were categorised in terms of the stage of the application development process and its substages.

In other words, the themes were analysed one by one looking for evidence on whether technology or routine was the more decisive factor and what kinds of reasoning were given for each of the themes. Another aspect was to also look for cases on the compatibility or incompatibility of routines and technology. Through factors like these, the intention was to find the areas where routines clearly played a role during the application development, and to further understand how context as defined within this research played a role during the overall process of application development.

4.6 Conclusion

Latour (1990) has noted that the macro structures of societies often resemble the micro ones in the sense that they are made of similar components and mentions that this applies especially to innovations, where a micro event can easily turn to a macro event by capturing all the micro events of the same category. This research aims to study application development from the perspectives of the relevant actors involved in the process, mainly application developers but also others, and to examine how the local context affects this process. However, despite our concentration on the so called micro level actors, it also aims to draw inferences for the more macro level factors that impact application development in a particular location.

Overall, the research has two major components, the technological one that comes from the hardware and software resources needed in the application development process and then on the other hand the context-determined routines that are constructed and given meaning by the relevant actors (and to some extent the researcher himself). Therefore this research adopts a critical realist ontology combined with an interpretivist epistemology, as this enables the researcher to combine the real components of technology that are ultimately based on physical laws and the meaning making that has been done by the individuals when they engage with the technology.

For the actual analysis, a case study approach is adopted which views the application development scene in Kampala as a typical case of a particular developing country region trying to set up its own application development sector. The start-ups working in the area are seen as embedded units of the case, where each start-up enables the research to see how well the themes found in one start-up match those in others. The data for the case was collected via interviews, non-participant observation and documents. The data was analysed using thematic

analysis and by trying to identify the underlying mechanisms that caused the themes and events that emerged from the data. In the following chapter, the case study findings will be presented.

5 Case Study Findings

This chapter presents the findings from the fieldwork conducted in Kampala, Uganda. It starts off by introducing the fieldwork location, both on a country and city level, and focuses on factors that can be seen as relevant for understanding the application development scene in Kampala. It then moves on to discuss the more concrete findings obtained from the collected data. These findings are categorised into the different stages of application development process that have been discussed in previous chapters.

5.1 Case Background

5.1.1 Introduction

Over the last decade or so, the region of East Africa has been often seen as a vibrant location for various types of innovation activities. The innovations have often been built around the mobile phones, but also other type of technology creation has taken place. This focus on mobile phones can largely be explained by the significant growth rates in mobile ownership accompanied by notions on the region leapfrogging technologies like landline telephones and general eagerness of adopting ICT driven technologies and devices (Ewing, Okolloh, & Rawlings, 2011). It is noteworthy though that there exists stark differences between the different East African countries regarding how much progress has been made in turning these societies into innovation hotspots. The general view is that the most promising area so far has been Nairobi in Kenya with its plans for the creation of Silicon Savannah, Africa's equivalent for Silicon Valley in the US with its high-tech companies and thriving businesses (Bright & Hruby, 2015; Kalan, 2014). However, other countries in the region have also expressed and taken steps towards creating similar enabling environments for technology innovation, and with that, application development, to occur. One of the main ways to promote the ICT sector has been the creation of technology hubs, which have been founded in the different countries and cities of East Africa. These hubs come in different forms, but can be mainly seen as co-working spaces for young technology entrepreneurs to set up their companies and to work on the technologies related to their business models. On a broader scale, the thinking has been that over time the technology entrepreneurs and start-ups would grow and transform the general business landscape in the region towards more ICT-driven industries (Ewing et al., 2011).

These views are to some extent in contrast with many of the developmental status indicators concerning the region. In most of these countries, general income levels remain low and there are challenges in areas like agriculture, education, health and security (African Development Bank, 2016; UNDP, 2016a). Corruption levels are also seen as problematic in addition to the general inequality levels in the societies, and many of the institutions remain weak. Despite

these factors and as the East African region has shown, just like the more developed nations also the poorer countries are equally willing to jump onto the ICT bandwagon and see it as a tool to promote economic growth and overall to create more developed societies.

Uganda can in this light be seen as a typical case of a developing country that in spite of many societal challenges aims for the creation of a thriving ICT sector. From a research perspective, the often stark contrasts between the developmental challenges and the modernizing notions of ICT make Uganda an interesting location for a case study on the role of context in application development. Within the confines of this research, the developmental status of the country should be seen as providing a particular context for technology creation as the intention is to *"focus on understanding technology 'in developing' countries"* (Brown & Grant, 2010, 96) instead of looking at the developmental role of technology in these places. The reasoning is that in a location where the developmental challenges are occasionally significant and certain aspects of the society have lagged behind the rest of the world, it makes it easier to see how the general context affects the development of technologies, and especially how these technologies and the surrounding context interact and fit together in the process of application development.

5.1.2 About Uganda

Uganda is situated in East Africa, and as shown in figure 5.1 shares borders with South Sudan, Democratic Republic of Congo, Rwanda, Tanzania and Kenya. As many other developing countries in the region, the Ugandan economy has grown considerably over the last decades. During the change of millennium, the average yearly growth rates were around 7%, from which they have declined and on average the growth rate has now been approximately 4.5% over the last five years or so. Among others, adverse weather, situation in South Sudan and availability of credit in the private sector have been seen as reasons for the smaller growth figures. However, the growth rate is expected to see moderate increase over the next couple of years as for example large infrastructure projects that have been largely financed by investments coming from China will be supporting the economy. The

Figure 5.1. Map of Uganda and Its Neighbours

value of public debt is around 36% of the total GDP (The World Bank 2016; Kynge 2014). Services make up of approximately 50% of the economy while industry and agriculture both hold a share of around 25%. The latter has been claimed to employ almost 70% of the population (Deloitte, 2016; Government of Uganda, 2017).

Despite the growing economy, the country still faces significant developmental challenges. The country ranks at 163 out of 188 countries according to Human Development Index (HDI), which makes it part of the category of countries labelled as "Low Human Development" (UNDP, 2017a). Of its population of 39 million, slightly over 33% are living in severe multidimensional poverty and another 21% considered being near this. The average life expectancy is just over 59 years and the Gross National Income per Capita is 1670 USD (2011 PPP$). Adult literacy rate is at 74%, and average years of schooling stands at 5.7 years. The average age of the country's population is very young, slightly below 16 years, and 16% of the population lives in urban areas (UNDP, 2017b). In its country report, World Health Organization mentions problems like poor basic infrastructure and weak public accountability, and in relation to health, shortage of qualified staff (World Health Organization, 2017).

However, over the last decades the situation overall has improved, or as the UNDP report states, *"between 1990 and 2015, Uganda's HDI value increased from 0.309 to 0.493, an increase of 59.5 percent[…], Uganda's life expectancy at birth increased by 14.2 years, mean years of schooling increased by 2.9 years and expected years of schooling increased by 4.3 years. Uganda's GNI per capita increased by about 120.0 percent between 1990 and 2015"* (UNDP, 2016b, 2).

In relation to ICT indicators, just like in many other parts of Sub-Saharan Africa also in Uganda the mobile subscriptions have increased over the past years and are now at 55% (2016). In relation to other indicators, 8% of the households have a computer (2016) and 25% of the individuals use the Internet (2016) (International Telecommunications Union, 2017). Approximately 5% of the people own smartphones (Pew Research Center, 2015). Furthermore, policies and plans of actions have been put in place to enhance the communications technology sector in the country in order to further the country's progress (Uganda Ministry of ICT, 2015). International organizations and multinational companies like Grameen Foundation and Google in co-operation with the telecom operator MTN Uganda have tried to promote application development in the area (Heim, 2009). Interestingly enough, Uganda has been ranked as the country with most entrepreneurs in the world in relative terms as 28% of adults in the country own a business or are co-owners of one. However, in most cases this is not out of choice, as there are not enough jobs to meet the number of people entering the job markets, further

calling into question how entrepreneurship is defined (Global Entrepreneurship Monitor, 2015; Patton, 2016).

5.1.3 Application Development Scene in Kampala

The population of Kampala, the capital of Uganda, is about 1.5 million (Uganda Bureau of Statistics, 2016)I. It is the location for much of the technology entrepreneurs in the country as well as for many relevant institutions that are needed to support application development. Among those institutions are technology hubs that function as office and meeting places for many of the technology and other start-ups. Regarding application development the two most important hubs are Hive Colab and Outbox, which differ in their operating model and to some extent in their general objectives yet share the intention to nurture tech entrepreneurs and star-ups within the area (Evans, 2014). Hive Colab is community based and originally free, but there were some indications that this had changed as one interviewee pointed out that the fee was approximately 150000 Ugandan shilling per month (just over 30 pounds at the time) (interview 35). The hub is interested more in projects and technology companies that aim to have a social impact and are not only about making profit. As a result, the focus of the hub is in areas like education, health and agriculture, and has worked to create transparency in its operations (interview 39). The impression was however that the hub was relatively flexible in offering companies and individuals working space even in some cases where the social impact was not entirely clear. Having said this, it is difficult to evaluate what constitutes as social impact, since it can also be measured for example in the ability to provide future employment.

Outbox on the other hand is to some extent backed up by companies like Google, Samsung and Deloitte, and also charged a fee from the entrepreneurs residing in the hub. It is more business driven in the sense that there was no requirement for the companies to create social impact. Overall, both hubs organise events, to which they invite individuals relevant to the field to make presentations and provide training in areas like app development and programming. In addition, at least in the case of Hive Colab, there was also collaboration with other institutions such as universities to provide training on how to develop business ideas further. On the surface, both hubs had similar characteristics, for example they both had a large open office space and tables where people were working on their projects. In addition, Hive Colab also had meeting rooms and smaller offices in addition to the open space. An important factor regarding the two was internet connection, as it appeared that the connection was better and more reliable in Outbox, possibly due to its more central location in the capital. When talking about the reasons why people had chosen one hub over the other, the better connection was sometimes mentioned. Also the profile of people occupying the spaces was sometimes seen as slightly different: Hive Colab was referred to as more ‘relaxed’ as well as community driven, whereas Outbox was

claimed of having more business-oriented residents. However, this does not mean that the people residing in Hive Colab would not have been interested in making profit, but more that the social impact objective, possibly also affected the general atmosphere in the hub (Fieldnotes 4).

In addition to the technology hubs, Kampala is home to many of the top educational institutions in the country, best example of this being the Makerere University. The university's College of Computing and Information Sciences (CoCIS) provides degrees in computer science and software engineering as well as business degrees from the university's business school. Many of the developers and start-up founders had a degree in computer science and in some cases from business administration or from a related field, and often those degrees had been obtained from Makerere University. The university's computer science department also had labs like Artificial Intelligence Lab, which worked in close cooperation with the Unicef's Pulse Lab in Kampala, which specialised in data science and its developmental usages. Also other universities in the city provided education in computer science and business, although there were some complaints that the education did not provide the start-ups with relevant skills in relation to programming and software development. As one interviewee noted, *"those skills you learn in the university were more directed towards people wanting to work in companies, but not so much for developing applications or teaching programming to a very advanced level"* (interview 34). In order to gain these skills, some people had taken additional courses in private education institutions located in the area (for example a private school called Uptech was mentioned by few) or alternatively by acquiring those skills over the Internet by using sites like Udacity.com or otherwise searching from the internet the information needed (interviews 29 and 35).

Another interesting factor in the city was the amount of international and local non-governmental organizations, some of which were active in organizing coding events like hackathons. These events usually had a social objective in terms of providing application-based solutions to problems in areas like health, agriculture or education. Sometimes hackathons were also organised by local telecom operators such as MTN Uganda. The prizes from the hackathons varied, for example the MTN hackathon gave the winning teams some funding, around 1000-3000 USD, to develop the applications further (Fieldnotes 3). The hackathons were sometimes organised in collaboration with the technology hubs or then ran individually by the organisations, for example the US embassy was involved in organising them (US Embassy in Uganda Hackathon, 2015). It was not entirely clear how successful these hackathons were in creating thriving start-ups, some examples of success cases did exist but also cases where the prize money was spent on something else or the start-up slowly ceased to exist after some time were plentiful.

In relation to start-up funding, the government had made promises on providing further funding opportunities for the start-ups but it appeared these promises had largely failed to materialise. Kampala did not appear to be a hotspot for angel investors either, although few companies had managed to raise some venture capital. In addition, plans to strengthen the information infrastructure had been made (Uganda Ministry of ICT, 2015), but it was unclear whether these plans were to become a reality. A foreign diplomat in the country for example realised having talked with a government official responsible for ICT that the official had not even read the report where the plan was introduced, and continued that *"this could be either of two reasons, one being that the official just had not done his job or that he already knew that the plan was never going to go forward"* (interview 45). Overall, internet connection varied greatly within the city, and was also subject to frequent power cuts that took place in different parts of the city.

While it was not entirely known how many of the start-ups failed a figure of 85% was mentioned, yet it was unclear where this information came from or whether this applied to start-ups in general in the region or only those residing in the hubs. Reasons for this was found from lack of money and training, but also it was mentioned that the place overall might have some impact, referring to the overall sometimes challenging environment in terms of lack of funding and developed institutions. The start-ups spend an average of two years in the hubs (interview 40), although there were also quite a few cases where this had been longer. Lack of investors was also often mentioned in the interviews, which meant that in many cases the developers and the members of start-ups either had other jobs that paid their living costs or were able to get money from other sources, which appeared in some cases to be the parents. Overall, with most of the developers having college degrees and some financial stability, it seemed that many of them came from relatively well-off backgrounds, although it was difficult to assess what exactly constituted as being well-off in the context of Kampala.

These factors provided a setting for the application development scene in Kampala. Naturally, other actors were also involved, but the ones mentioned above were the perspectives that the researcher himself observed during his fieldwork and those also emerged from the interviews. In the following, the data collected regarding the application development process will be shown in a manner that corresponds with the usual steps of application development.

5.2 Application Development Process

5.2.1 Ideas for the Applications

As it can be expected, there was a wide range of ideas regarding the different applications and technologies that the starting entrepreneurs, as well as the already more established ones, wanted to develop. In addition to software, hardware innovations were also developed, such as

drones aimed at killing mosquitoes and devices to measure sweat to know ovulation cycles. In some cases it was more about finding the right hardware instead of inventing it, like inexpensive screens that could be used to advertise in the local busses. Some applications were clearly web-based products, for example a Twitter-type of application for schools, or a site that would publish information about government and increase transparency. However, most of these were in the idea stage and had not been even built yet, or then no one had yet thought about how to fund the projects or how to make them financially viable (Fieldnotes 8). The origins of ideas presented below were taken from start-ups that had already developed or started to develop the applications.

5.2.1.1 Origins of Ideas

Overall, the origins of the ideas for the applications could be traced back to the location of the developers, i.e. the local context. In some cases, a particular event had led to an idea that functioned as a basis for the application. As one start-up owner explained:

> *"So what happened was that there was a fire in my brother's school, and someone called the firemen to come there, but it took them for a long time to reach the place. Then, um, people were asking what took them so long, as the fire had gotten worse in the meantime, and they [the fire brigade] said that they had difficulties in finding the place, cause there are not addresses on every street, and they did not know where to go, and also the roads can be tricky cause you might end up to a street that is a dead-end. So I started thinking a way to have like a code for every place, and that is when I came up with a thought that every place, like every small area, could have a code that would refer to a specific spot"* (Interview 28).

> *"So my younger brother was in this school, and then there were other kids, whose schools weren't that great, and I was thinking is there a way for example to share some of the knowledge across the schools and disclose the difference where the teachers are less good. And I went to them, the students, and said let's try to make a way to share work with other students, and if you want to be the best it's good to share, and I was like, let's see if I can figure this out."* (Interview 19)

Not all the cases were based on particular situations only, but more to occurring phenomena or events in the society that a developer or a start-up partner had noticed:

> *"The idea we had in mind was that there was so many people passing out from cervical cancer within our society, so I came up with an idea for an application to help people to test cervical cancer".* (Interview 23)

> *"There are a lot of people spending too much money here in Africa, but so many times the recipients of this money always disappoint the owners of the money. I give you an instance, someone in the west or somewhere, I'm gonna send [him] for example five million shillings to buy a piece of land, or take [their] daughter to school, but […] once they receive the money [..] they have the flexibility to do anything with that money, you understand, cause the owner of the money is very far away. So we say 'why don't we create a way to transfer money from one part to another, and bind that money to a*

purpose, so we give you a coupon which is an equivalent to the value that you give out'" (Interview 7).

Also what occurred was that people had worked in something similar before, which led them to create a programme that would meet a perceived need in some other relevant area:

> *"Previously I worked with the biggest ERP [enterprise resource planning system] for hospitals which is used in East Africa […], I worked as a business development manager for about two years there, and I realised that there's a gap for that in schools, and no one was doing really anything comprehensive like that for schools, and I wrote some ideas how it should be, and that is how it got started"* (Interview 14)

In general, the ideas stemmed from personal experiences, professional or otherwise, and those led to an intention to create software or an application around the ideas. All of the interviewed seem to trace the origins of the ideas to the local context. However, when discussing the works of others, especially regarding applications that had not been successful, those were seen as copies from something developed elsewhere.

> *"There have not been many successful apps, but the thing is they don't often actually solve any real problems. People build apps to win a hackathon, but that's where it stays: when they win the money it doesn't go further. And regarding areas like e-commerce, those applications are not really Ugandan, they come from somewhere else. But you don't find a lot of local successful apps, and often they just try to mimic some other applications build elsewhere."* (Interview 9)

This statement that some of the applications borrowed from the functioning of applications built elsewhere or just tapped into the APIs provided by external actors seemed correct. However, when explained by the persons who had invented the ideas, the core of the ideas for the applications stem from the local context, either from an individual or reoccurring event or from a phenomena that was pertinent in that particular location.

5.2.1.2 Other Factors Affecting the Ideas and Content of the Applications

> *"So everything is contextual, depending on the market you're trying to approach, and the contacts or the environment of the users decides whether you're going to make it or fail".* (Interview 31)

Contextual factors refer to the social, cultural and other factors that were present in the location or contextual setting surrounding the developers and their targeted users. Those also affected the ways companies directed their actions and developed technology. For example, one interviewee noted how these kinds of contextual factors linked back to the building of the application:

> *"There is quite a bit of hierarchy in schools and the head or the director has to be able to confirm yes, these are our students, but not the teacher, because they are more in charge of the academic part. Also a good thing is [that] you can use the same version for*

primary, secondary and all levels of school, just change few things like the subjects." (Interview 14)

Also cultural norms affected the reasoning behind the applications, and were among the key reasons why certain applications were developed.

> *"Yeah, so you ask a friend that she went to the doctor, and then it's all about [how] they asked me to open my legs, little bit too uncomfortable […] Because women are shy, they're not comfortable going and checking themselves, and the fact that they are not seeing the symptoms make them go like 'I'm fine', or it just comes for two days and they ignore it, they don't know that the longer it stays there more dangerous it becomes. So the way we made it is that a person can check herself by using this hardware and software."* (Interview 23)

In certain ways, this was similar to the note on the origins of the ideas of the applications, but in these cases, the local norms and rules had a more direct role in deciding how the application should be built. Furthermore, these contextual factors had an effect on the type of technology that had to be used in the applications.

> *"The good thing with the cooperatives is they bring the farmers together, for the cooperatives they buy inputs in bulk, distribute [those] among the farmers, they organise the trading, it's like a personal relationship and for us it's a way to reach a big number of farmers especially if they're small farmers, with the big ones you can deal with them directly […] We have a niche, because we also serve the small farmers as we offer SMS and stuff like that."* (Interview 21)

In addition, location also affected how certain factors were defined, which in some cases was quite different for example from the ones of Western countries.

> *"Then we struggled with [the] quality [of agricultural products], we spent about six months trying to nail down the problem of quality, like in the European market quality is defined like the size of the fruit but here,[...] sometimes the buyers would say yes I care about the quality, but when you go deep down to understand what their definition of quality is in the practical sense, you realise that it is very different from what they actually say, so it turned out that if they are buying maize, they don't care about what is the seed size or what is the moisture. They have something that they call the average quality, they don't have the term average quality but in their marketing negotiations, they have something that in English would be average quality, and it is almost like a standard that has been established within the agricultural trade market, that someone buying maize from a farmer knows that I spent maybe 0.1 percent on stones, meaning they do a bit of sampling of this maize, and they look at it and say, bit of stones, not too much, and of course there are thresholds, but those thresholds cannot be formally defined, because they don't have a great machinery to do that".* (Interview 18)

Interestingly, certain contextual factors made it also more difficult for the companies to succeed, as the general perception of locally built products and services was that those were often of inferior quality.

> *"We have a mentality issue, not only in Uganda but in Africa, even me, I go out and try to buy something, and then I go like 'you know what, I'm going to buy it on Amazon', because I don't trust what I am going to buy from my local shop here. So if you're going to build something, make sure it is top notch, and that it doesn't crash, otherwise you're just going to give an excuse like 'you see this is why I don't use local apps' and they will just go and use something else. And it's the mentality of the people like what is foreign, it's much better than what is local, that's the thing that has been going on for a while. There's that and then, well it's a sort of racism, it also happens that, even when I walk into a boardroom with you, my peers, at least the older generation, is going to trust me more because I am with a white man, and I think that is one of the biggest problems we have in Africa."* (Interview 24)

Overall, these contextual factors not only affected the ideas of the applications, but also the more non-technical areas of application development like business management. One area where this had a clear impact, on both the types of applications that were built and also the way the whole start-up itself was managed, was related to the monetizing aspect of the applications, and with that, the financing of these start-ups.

5.2.1.3 Funding and Monetization

Overall, funding in Uganda for start-ups was generally seen as scarce, although some exceptions of start-ups that had obtained some funding did exist. In addition, comments were made that investing in start-ups was not necessarily seen as the best way to use one's money. This also affected the long-term planning of the start-ups.

> *"In truth, every start-up in Africa, unless you're in South Africa, or Nigeria, or maybe Egypt, you know the big economies, all of us have the same issue which is funding, we don't have an investment culture, so like in Uganda people will raise you 100 million shillings for a wedding but they will not raise you 100 million for a business [...] And also, building first a large user base like Twitter, you see that wouldn't work because you just cannot sustain that, and like the Silicon Valley thing that you have an idea, get a family round, raise some money, build an MVP [minimum viable product], stop there, go to an investor who is going to give you money to run it for 2 years without necessary having to make profit, and that just is not possible here".* (Interview 24)

> *"Funding here is so bad, and that is actually delaying the roll-out, cause now that we are patenting, we have to pay the fees, we have to import the hardware… I wish the government would consider start-ups and fund them to get out to the ecosystem […] The situation is better for example in Nairobi, Kenya, where you can get funding from different sources like the first lady of Kenya, and I was thinking why don't you have this here, where the only way to get access to money is through a challenge, or a competition, and in those, if you don't have a properly working concept you won't stand a change."* (Interview 23)

However, as noted above, some companies had been more successful in raising capital.

> *"But we have been able to get quite a lot of funding, so in that sense we are ok for now, we have quite a bit of money to develop this, even though we are in the process of trying to collect some more money as well".* (Interview 10)

This was not common however, and as a result, many of the developers had to work for other companies while developing their applications.

> *"You get so many additional expenses, so […] you need incomes all the time, and right now we don't have them too much. I also work for the hub, so it kinda enables me to at the same time do what I like, I mean having the start-up, while working with the application."* (Interview 4)

The lack of funding meant that companies had to focus on an application idea that at least in theory had the possibility to provide incomes right away. This led to a situation where many of the start-ups developed applications around existing flows of money and were eager to show for example how their application enabled people to either save or make money.

> *"We were thinking about 500 dollars for the licence, because it is useful for the communities and helps to them to make money as well".* (Interview 17)

> *"This is good also because people are already paying for airtime, I mean there is money there if you can tap into that, and since this brings savings, you can, and people are accustomed to pay for airtime."* (Interview 3)

The need to monetize right away placed restrictions on who you could sell, and also on the sales agreements one could make:

> *"There is also a limit to who to sell, if someone is very poor, there is no way to sell, does not matter what you do, they just do not have the money. Like if someone makes 20000 shillings a month, they don't use it to something like this. […] Another thing when selling for the poor is that money as a concept that measures everything does not mean much, I mean if I say this saves you time that kind of statement does not really make much sense to them, they are like it does not really matter. And also how you collect the money, has to be cash with the poor, it is not like there are credit cards. Although, mobile money might work, but it is not that the poorer have even that necessarily".* (Interview 30)

As also hinted in the comment above, lack of certain technological infrastructures made it sometimes more difficult for the companies to collect money.

> *"These guys pay us like a weekly subscription fee, that's the way we monetize, we cannot take a cut like Uber, because we don't have integrated payments here, and also because you have this online offline offering, where you can take the bodas [motorcycle taxis] from the street which is something you can't do with Uber […] There is a lot of money to be made with regular people though, like here with the bodas, million dollars changes hands every day, so it's not a small market."* (Interview 10)

Regarding the comment above, the possibility to integrate payments into the systems became possible at a later stage, so the situation got better in that sense (Fieldnotes 10). However, funding remained a problem, and affected what types of applications were built.

> *"Well there is also the money […] it is easier to start from the top and going down, than the other way around, so if you target all the boda guys [motorcycle taxis] in Kampala, but you realise they are a bit stingy with the money, but if you can get a solution for the*

guy in the top, who has the money, you can find yourself a way of working down". (Interview 12)

"Most people that operate in the region give more financing to the social businesses than pure business, so you have to be doing something in education or something like that to get funding, but if you're doing something for example regarding travel it becomes more difficult […] most of the money comes from philanthropic organisations, to create impact". (Interview 40)

Some also saw non-governmental organizations (NGOs) operating in the area as possible sources for funding, although it was not always straightforward to get these organizations to finance the start-ups.

"One way is to work together with the NGOs, but it can be difficult to get them interested, you have to know what they want, and it helps a lot if you have an insider who can tell you what they want, and then you go and tell them what they want to hear, and they can then give you some money, and then you can do the things you wanted to do, but you need them first to get the funding". (Interview 17)

In sum, the original ideas for the applications and businesses stem from the occurrences and routines in the local environment, which were further shaped by the existing cultural norms and other contextual factors. Furthermore, other issues like the lack of funding also impacted the choosing of the ideas for the applications, as every idea had to have some connection to making money, be that in terms of generating revenue or receiving funding from some source.

5.2.2 Refinement of Ideas

Although ideas were constantly refined throughout the development of the applications, this occurred first in a phase of further trying to verify the accuracy of the original ideas. In some cases this was done after the first versions of the applications had already been launched, and it appeared to come as a bit of a surprise for the start-ups that the original assessment of the idea, which the applications tried to capture, was not entirely correct.

5.2.2.1 Perception Verification

In general, many of the application developers realised that there was not necessarily a need or want for the application the same way they had predicted. However, most of them were aware of the importance of understanding the users' views.

"I just have been asking myself two things: am I still serving a real problem, addressing a real need, and the second one has been how I can make it sustainable. So you just more like adjust, you add few things and take out some others, like one that really does not make sense in my case [is market prices,] my app doesn't have market prices, cause I know I'm not adding value to anyone with market prices" (Interview 31)

"I think there is potential, but it has to be for something that app users are not going to get elsewhere. If you are going to build a WhatsApp, you are wasting your time, same

thing with another Facebook, but if you are building something that people are able to use and uniquely fits into the market, that makes sense". (Interview 24)

When discussing with developers on the other applications that were built, what often surfaced was that it was seen that most applications were not actually solving a real problem. Furthermore, sometimes it also turned out that the original idea was not simply technologically or otherwise possible, and the original problem had to be redefined.

> *"The problem was that once we got some consulting on the issue, we realised that it takes two to three weeks to test cervical cancer […] so after getting all that information, this consultant advised us to look at more the long term causes of cervical cancer, and that's how we came up with the idea of BBKit, and it's based on seeing when there's a lot of unhealthy bacteria in the vagina".* (Interview 23)

In addition, another realisation was how many other additional factors had to be taken into account, and how the technological development was not enough in itself. Furthermore, other factors like the constraints put in place by the technology or the lack of technology started to surface.

> *"One thing we have realised is that we need that strong operational [base] built[…], so we cannot be like Uber that has everything online, we need to be a little bit more hands on regarding driver management, like they should be having their reflective jackets all the time on, we would also like to microfinance phones to the drivers, so they get to use WhatsApp and so on and of course our app. So right now it is even more important to get the operations working than the app, and for the app we need to train the drivers as well, how to login, how to buy a bundle and stuff like that, and it is all something that Uber doesn't have to do."* (Interview 10)

These other factors meant in most cases more manual work for the start-ups. It also turned out that people also understood certain factors differently even within the same location, and this was something that the start-ups had not necessarily foreseen.

> *"It would be helpful for example if the application had a map for them to show where they need to go, although sometimes they don't really understand the point of a map, they prefer calling and asking us to guide them there over the telephone."* (Interview 4)

Also other, possibly problematic issues related to usage became clearer.

> *"The problem is that people might still bypass the system, because once they know the person doing the laundry, they might just take the persons details directly and not use the application again."* (Interview 3)

On a more positive note, some initial challenges were solved as the start-ups worked more on the idea.

> *"Now we also take a cut when they're selling their commodities. So the starter package, you get the basic functionalities, you can trade commodities, you can buy them you can look at the market prices, and basically do everything you want. But in case you want to do extra things, you go for the premium package."* (Interview 21)

Overall, the verification of the original ideas enabled the start-ups to move closer to the views held by the targeted users, which also enabled the developers to better understand the problem they were trying to address as well as the feasibility of their solution. Interestingly enough, there were also cases where the idea itself seemed to work, but the reasons why it worked appeared to have been misunderstood by the start-ups.

5.2.2.2 Misunderstandings Regarding the Idea

Occasionally the problem that the start-ups tried to solve with their applications seemed obvious, like providing access to safer motorcycle taxis. However, it then later turned out that the reason why people wanted to use the system was that it just produced a continuum for the existing way of performing a particular action, without actually changing the performance of that action in itself.

> *"We thought all those helmets would be important, but the Ugandans are like 'I never really wear a helmet', it's not part of the culture, we're trying to change that of course, that is is important to wear a helmet, we have hair nets that they can put underneath, so yeah we're doing that."* (Interview 10)

Also, the reasons why people used an application had additional consequences, either positive or negative, which to some extent affected the usage of the application.

> *"Like in a restaurant, I am now developing this point of sales system and the waiters do not use it, because if everything is in the system, they cannot fraud, or it is more difficult. And the manager does not want to upset his workers, he wants to be in good terms with them, so he does not want to force it too much. And I thought he wanted the system to keep track of things, but it seems his first interest was getting printed receipts, I mean instead of writing them by hand, he sees that when a receipt is printed and given to the customer it looks more professional. And he was very interested in having the receipts printed in the kitchen, so I had to also think of these things how to connect things, like from the bar to the kitchen."* (Interview 8)

It was not always clear for the developer why certain functionality was needed, as highlighted in the comment above, where the developer was not really sure why there was a need to have the receipts printed in the kitchen. Also in some cases technologies were wanted for the sake of appearance, without actually taking the full benefit of their potential.

> *"For the businesses here, the web is more like a window, or the website, it is good to have one but they do not use it in any way, like there are only some who want to have an online store, and even if yes, then without any payment [options], so it is just like a catalogue, but you still have to go to the shop [to buy]. It is like they don't really trust it, they just want to have one, cause that gives them something, like they seem more modern."* (Interview 35)

In addition, other factors, which had not been necessarily thought about by the start-ups, had sometimes an important role in deciding whether the targeted users would actually use the system. Many of these linked back to contextual factors.

> *"The sellers want someone who will offer them a better price, but they are also interested in someone they can rely on, someone who says I'm coming to buy next week and they really come next week, or they are coming today and they come today. The farmers and also the sellers have immediate needs, like I planted this maize and by the time the school term starts I will be ready to sell, so the reputation of the buyer was very important for the sellers and that lead us building a reputation mechanism."* (Interview 18)

5.2.2.3 Technological Constraints

Another factor that caused the developers to rethink their ideas was reasons that stem from the technological factors. These issues were not so much about having the skills to use the available technologies, but more related to the other attributes of the technology, such as price and their prevalence within the targeted users.

> *"The Internet bundles are still way too expensive, so that is an additional blocker for us, you need data to stay connected and use the app."* (Interview 31)

> *"This is Africa, the agents who are supposed to operate in this market have feature phones, so you don't have the luxury of the options that come with a fully blown web interface software, so we took a lot of time and started doing it as an SMS interface application."* (Interview 18)

Even in the case the technology was available in the right form, it was sometimes difficult to use or clashed with things like the working patterns of the people who were supposed to use the application.

> *"There are also other things, like we had to ask the collectors to go [to] these places at a certain time, and they prefer to go [to] these places when there is no traffic, and then it might be that at that time the app does not work properly […] Also it [the device] is bulky, you carry things with wires and laptop, and you need two people to operate it, and also the battery runs out quite quickly."* (Interview 16)

In most cases, the lack of a certain type of technology meant that a different type of technological approach had to be taken, such as building an application for basic phones instead of smartphones. Another way to solve the issues was to think about ways to provide the targeted users the technology.

> *"We are thinking about ways to get the users that we need for the application, they are very poor, [and we are trying] to think of ways to give them smartphones. One option could be if we could get the local operator to give us some basic smartphones that we could then somehow give to the users. That would make our job easier because now we have to be constantly available in case they call. […] It would be just easier if you can automate things, and do less of that manual work like being on standby and all the time calling and answering calls and so on".* (Interview 3)

As the comment above shows, not being able to use the optimal technologies meant more manual work, and in general complicated the functioning of the start-up.

> *"We haven't found a company that would do that [payment integration], there is a company here called, what was it, Yo Yo something, they have a payment, like a wallet type function you're able to use. That would be great to have it, that's a proper way to monetize instead of asking drivers money every week, it would be just so much easier, and that has been a worry for the investors as well. I mean the potential is huge, there is like 80,000 motorcycle taxis, that is like 5000 more than taxis in New York, it's insane, but how do you monetize this, that's the challenge".* (Interview 10)

Lack of standards or compatible technologies was also mentioned in few instances as examples of technological factors that made the development of the application more difficult.

> *"It got a lot more challenging as we scaled up to other countries, because that meant we also had to incorporate different phone models to the system, and they all used different kinds of cables or even configurations to transfer the data to the software. And you also have models that we had never even heard of, or the cables themselves could be counterfeits which meant they were lacking a pin that they should have and therefore did not work at all, and we had to come up with a way to make them work".* (Interview 2)

However, one way to overcome these issues was to basically hope that the problem would solve itself within the near future, and quite a few developers took this approach for example regarding the smartphone penetration in the country. Also some commented how the habits of people using technology were changing across different age cohorts and social groups.

> *"I wouldn't build an app that's based on SMS, […] people will catch up [having smartphones]. So like five-ten years and we are there, because I think by then people will stop more worrying about the phone and it will be more about the service."* (Interview 40)

> *"The other thing with technologies [is] that it will never go back, the numbers will only improve. […] You're not going to go over full blown to whole Uganda in one year, you're probably gonna start off from Kampala, and then maybe Entebbe, and let's say in three years people will be having smartphones all over Uganda."* (Interview 11)

> *"There will even be an app where they are tracking whatever happens around the country regarding the elections, and also media communication, the media has a team on social media, and all of these are making stories […] people were saying these things are for young people only, […] but more people have started to join, Twitter, more Ugandans are using Internet, smartphones, so they can get real-time information."* (Interview 29)

At the same time, others opted for different types of technologies or data that were available and more typical in the location.

> **"***There is a need to harness new type of big data in the sense that lot of the traditional data is not so easily available, and for example the type of roofs a building has can give insights on the income level of the people who live in the house, and this can be seen through satellite images. The type of data like Twitter feeds that is used elsewhere is not so much available in Uganda, as not too many people use it."* (Interview 46)

Needless to say, these technological constraints had important implications for how eventually the applications were built in terms of choosing the proper platforms and programming languages.

5.2.3 Choosing the Technological Basis for the Applications

What is noteworthy and highlighted in the comments, is that the evaluations of the appropriateness of different technologies were often quite subjective, and different developers had differing opinions on the same technologies. This could partly be explained by the different user groups the developers were targeting, but in some cases it also appeared that not much research had gone into studying the situation. Instead, these perceptions were more based on general estimations that the developers had on the topic. This point was highlighted even further in cases where the developers themselves were the ones who had gotten the idea for the applications.

5.2.3.1 Access to Developers

Once the idea for software or for an application had crystalized, the start-ups needed someone who could actually build the application. In most cases the developers themselves were the ones who had formulated the idea, and in this sense it was easier for them to start developing the applications or software, as they could do that largely themselves and were also familiar with the technological tools and resources.

> *"We are using Drupal [content management system], which my co-partner is familiar with, and most of the modules seem to be there already. As I told you I've been working on closed models, and then we had to build all these modules, so the good thing is that as a company we have been able to use open source from day one. Some very tiny bits we have had to customise."* (Interview 14)

However, in some cases the development of the application took a little bit of learning first, or due to the small number of people involved, not everything could be built at all.

> *"For the offline part I really had to learn, there were a lot of things I had never even heard of and I had to learn those."* (Interview 19)

> *"That [Android operating system] has the highest number of people in Kampala right now for smartphones […] The app itself was very quick to build, it took us less than 72 hours. The way we learned Android was that we did few courses in Udacity [online course platform], and that way we got the basics."* (Interview 3)

> *"Well the number of Android users among our targeted users was very limited when we started, but most of the people we targeted had access to the web."* (Interview 21)

When the person who had gotten the idea was not a developer, there were sometimes difficulties in finding people who knew how to code and develop the software.

> *"I struggled to find someone to build this as I don't come from a programming background, I come from a law school, and of course it's very hard to get someone to develop this as you don't have money, and that was my challenge in the beginning. And some of the guys were really obnoxious, asking ridiculous amounts [of money] for their work. But finally I met someone who could help me with the programming, he had some ideas and I had mine so we spent time sketching what the program would look like."* (Interview 19)

Also finding talent was not always easy, especially if something else than pure coding skills were needed, even in cases there would have been money to pay salaries.

> *"Right now we're trying to find a tech lead, […] the graduates here are strong but we need someone with co-founder abilities and [to] take ownership, someone who manages multiple technologies, and those can be difficult to find."* (Interview 10)

Here it is difficult to say how many ideas were never turned into applications because of not having access to developers or not possessing the skills to build the applications. However, in most cases the start-ups had been established by developers, and as noted, the problem of building the applications from a skills perspective was less of an issue. This sometimes also meant that the application developers realised the importance of other factors that affected the development of the applications.

5.2.3.2 Choosing the Technologies

The start-ups were trying to balance between technologies they were familiar with or able to learn relatively quickly and made most sense in relation to the targeted users and the ones that enabled them to build the type of application they wanted.

> *"We have thought about maybe building it for a basic or feature phone but the user experience would just be lousy."* (Interview 31)

> *"We had to do the iOS because some of the traders are using that, but the market for example for Windows is very small, so we will do iOS and get Android back, and then of course SMS and USSD, but I don't think we're going to be focusing on that because the technology is changing very rapidly, so the SMS we're just using it for the time being as a stepping stone."* (Interview 21)

Also in the background there seemed to be some sense of what was in fashion and easy to use, and for example it appeared that the developers themselves were especially interested in building mobile applications because of the general attention these technologies had attracted.

> *"I am not even sure if I really need a laptop for anything, I think my mobile lets me do most of the stuff I need, so it's like why waste money on that, my mobile and all those apps can take care of most of the stuff, or actually, practically everything."* (Interview 55)

> *"All the developers, nobody cares about desktop development, or desktop apps, everybody just wants to develop mobile apps, or do web development. This stuff [desktop applications] is not done by many of the developers here. To reach students,*

you cannot reach them with desktop applications, and that is one thing why Android app can be better." (Interview 8)

However, it was relatively clear that an application that worked on SMS only would also make it possible to reach larger crowds, and not everybody were either so sure that things would change rapidly in terms of factors like smartphone prevalence or people's willingness to use the applications in comparison to technologies like SMS or voice.

> *"People still use it [SMS], especially banks sending balance notes, or reciting inquiries, it's still there, although now people are going for voice, especially schools. For example if a principal of a school wants to call parents, he records a message to all parents and he'll make sure that all parents will listen the message as opposed to [text] message, you don't know if it was received, you don't know [..] Currently I don't really see so many changes regarding that [increasing use of smartphones] in the near future."* (Interview 37)

Furthermore, in some cases there was also importance in choosing a technology that enabled the developers to build something quickly. This was due to the objective of testing the application in the market or the whole concept overall, and in those cases adding complexity was reserved for later releases and updates. In addition, the application was sometimes designed for particular audiences, and the other clients were hoped to be reached in some other ways.

> *"So using the app, that's usually the foreigners, and maybe some young Ugandans, but it's been a bit of a black box because it's been difficult to get feedback on the app because it's such a simple app, we don't actually have user data. We need to get better at this and we have a second version coming in, hopefully with that we can kinda see who signed up and everything. We also created a hotline, you can call the hotline if you want a driver so we connect you with a driver, like instead of using the app, which is brilliant because you can optimise access [to the service] and also get data. So far like 99% of our customers are women who are calling in, they don't have the app or where they are there is no coverage. So that was our mistake but it was easier just to design a very simple app, but that also means we don't get a lot of feedback."* (Interview 10)

5.2.3.3 Knowhow and Skills

> *"I chose Python, I did not use the raw Python language but the Django framework. Also Java, I am very comfortable with Java, I have written software for smartcards using Javacard, but when I look at Python, you can quickly prototype, and see if something is going to work or not, which means that your turnaround on a given programme is much shorter than with Java, then also the extensive amount of libraries […] so [the application] on the surface is very simple, but there are a lot of machine learning techniques under the hood, so you get the power of all these elements in Python […] the developing world, Africa, Uganda, is not short of talent, but experienced people to direct this talent."* (Interview 18)

Skills regarding particular technologies were not seen as an issue by the developers themselves, although some isolated comments were made regarding the skills of other people. Overall, most

saw that the skills base was there, and it was mainly other than technological skills that were lacking.

> *"The technology is not difficult, you can write the program you want. I don't think I have never really had problems like not being able to do what I want, and you can always ask. It is the other things that make it difficult, like how to get people to use it [the application]."* (Interview 8)

Some references were made however that the universities and educational institutions were not always able to provide the kinds of technological skills that could be used, or that the skills learned at the university were not always helpful for the developers in relation to their work in the start-ups.

> *"The hubs are excellent in giving the young students skills and to turn them into something practical, but also for those practical skills they need to be in the university. So the start-up should not be a place for picking up the skills, ideally the universities should be able to provide these necessary skills, like learning new programming languages, version control and things like that."* (Interview 18)

> *"Here it is ok [finding programmers] but the sort of talent I'm looking at you won't find it here. For example I would like to have a doppler signal processing engineer, but there is no course that I know of that teaches that type of signal processing engineering, you just touch the topic a bit but it's not a degree."* (Interview 24)

Sometimes help regarding certain areas was also sought from external institutions, although this was less common among start-ups whose products were purely software-based.

> *"We are getting some advice from UNFPA [United Nations Population Fund], and MIT [Massachusetts Institute of Technology] was actually just replying to one of our mails and asking us to meet up and give advice on how we can improve, like how to make the measurement more accurate. Right now we are around 77 percent, we should get at least to a point where we are 95 percent accurate."* (Interview 23)

In sum, almost none of the interviewed start-ups said that they had lacked the skills of building the software, or if that had been the case, they had been able to learn the needed technological skills. Only in very specific cases was there a clear lack of talent, which tended to be quite specialised, or then the skills were lacking in the non-technological areas like operational factors of running a start-up, which was sometimes seen as the biggest problem.

> *"The problem with the local start-ups is that they have always been started by a developer and nobody has tried to study the market, nobody's thinking like this is the industry I am focusing on, how big this industry is, and how am I approaching to get a big market share. Instead they're looking things like Amazon and seeing they make money, and they're not getting the fundamentals. The game in Africa is completely different, and that's the thing, they just look at those Facebooks and those guys have different stories, this is Africa, and here the stories are different, you have to learn how to bribe, immediately everybody has to be your friend, and none of these start-ups try to*

learn these logistics first, they just deploy whatever they deploy but by default you have to know which market you're entering." (Interview 20)

As pointed above, institutional factors also affected what types of applications were developed and how, and in some cases they also had a direct linkage to the used technologies.

5.2.3.4 Institutional Factors Affecting Application Development

One factor which possibly guided the types of applications developers ended up building were the hubs where many of the start-ups were located. Although open to many different types of tech start-ups, at least one hub had a clear agenda of trying to promote businesses with a social goal in addition to making profit, whereas another one was more concentrated on start-ups whose sole purpose was to make profit.

> *"The other thing is that they have also branded themselves as a hub that attends to social issues, so I'm not sure what that is, is that like regarding funding or to support the entrepreneurs towards that direction, as for us, no matter what you're doing or who you're attending, in the end of the day you have to make profit."* (Interview 40)

Also NGOs and development organisations were seen by some as promoting the development of certain types of applications, and occasionally similar applications and systems were built by different organisations.

> *"It's like all the NGOS, they get funding for a project X, and they execute project X, and sometimes several similar systems are created at the same time, and it can be that the projects themselves make sense, but in the end, it is just waste of resources I think."* (Interview 22)

The criticism that surfaced in some of the interviews was that it was not clear whether all these different kinds of applications and approaches promoted by these organisations were really needed. To some extent, the same also applied to the technological tools that were provided by foreign corporations and sometimes also actively promoted in the hubs.

> *"I think data science is more like an UN idea, and actually the rise of data in Uganda is mostly been supercharged by NGOs, like bit more openness, more transparency […] And also sometimes I feel like you are pushing a technology to me, like Facebook has open sourced most of its APIs, Google does it a bit, but the point is that they are pushing these tools into my hands but I don't have where to use it for, I don't have big data, I have small data, this tool is not going to be efficient."* (Interview 44)

Although many made the point that especially Android smartphones were slowly gaining ground, it was clear that one of the aims for the international corporations to organise events, and to a more limited extent, provide training and funding to start-ups was linked to their own aspirations to gain market share. In some cases however, this was not necessarily the best approach in relation to the overall market or to the types of applications that were built, as the

claim often went that the applications failed to solve an existing problem or meet a real need or want among the targeted users.

> *"The application is now available on Windows platform […,] the reason why we build the application on Windows at first was because the competition was organised by Microsoft."* (Interview 23)

> *"Companies such as Facebook, Google, we like them but they don't always offer the best solutions, so when they come around, there's a lot of buzz, there's a lot of music, there's a lot of cookies, and a lot of software written too but most of this software is not solving immediate problems, or problems that have been provided by the environment. […] Indeed like instead of having such hackathons for developing Android apps, the thing is [that] people are smart, they could just as well watch Google videos, they can learn to develop Android on their free time [so there is no real need for those hackathons], but if Google turned around and instead sponsored three professors to stay in Uganda, those professors do not have to teach Ugandan student how to write Android, they just need to present them with a problem, that we have this problem then we need to solve, could Android be the platform, could USSD be the platform, could SMS be the platform, and if they did that, great ideas could be delivered to people."* (Interview 18)

Another example where institutions like embassies, NGOs or international aid organisations played a more visible role was through organizing different types of hackathons. These hackathons often had a particular theme like health, gender issues or education, and through these themes these entities established the type of applications that were to be built during the event. As noted, many of the winners of these hackathons gave up developing the application relatively soon after, although there were also examples where an idea that had been started in a hackathon had been turned into an actual company. However, this required additional resources on top of the price money won from the hackathon that for example the tech hubs tried to provide.

> *"I organised a start-up weekend hackathon and that is when I realised that we need much more, cause I realised it that in my events, practically that is where it ended [for the start-ups], they did something there and that was it […] so then I decided to set up a structure that could help people like that, like taking in people and giving them access to resources that can help them [to] become better and as a result we started the hub."* (Interview 40)

Being able to partner with international organisations or bigger corporations was not necessarily seen as a bad thing as such, as it could for example be seen as proof of quality and something of a sign of approval. This was especially needed in a location like Uganda, where the perceptions of locally developed applications tended to be quite low.

> *"We have been aligning ourselves for the past three or four years with NGOs [working] in maternal health, maternal newborn health from UNICEF, WorldVision, Save the Children to UNFPA [United Nations Population Fund], so all these are aware of our existence. [..] And for us, that would be a stamp of approval, if UNFPA is your client, that is an approval also to medical doctors, and it is easier to sell the product."* (Interview 24)

To balance the impact of international corporations, sometimes it was hoped that the government would step in and help, especially when the impact of these organisations was seen in less positive light.

> *"One of the operators, when it was bought by this Indian company, the new owners thought that their competition are value added service providers and coming from India where they innovate a lot, they thought they would do everything themselves, so they closed doors to all local content providers and they left only institutions like banks or government agencies. And that's where government comes in, with proper legislation they should not have allowed that to happen because it stifles innovation."* (Interview 18)

Overall, institutional factors had a role in shaping the ideas that were behind the applications as well as how they were built in terms of the technology. In a country like Uganda, some of these institutions or organisations were foreign, especially the big corporations and international organisations, and although in some sense forming part of the local context, their agendas appeared to come largely from outside the country.

5.2.4 Building the Applications

After having an idea for an application and choosing the technological basis for the applications, the development of applications followed. As mentioned, the ideation stage and the actual building of the applications often happened simultaneously, although the two stages formed to some extent rather separate processes as such. Overall application development should be understood as an ongoing process, where updates keep on reoccurring, yet for the sake of clarity the specific issues related to the actual building of the applications are discussed here separately.

5.2.4.1 Location

Many of the applications were being developed in the hubs, which enabled the start-ups to cut costs as the hubs were relatively inexpensive (and sometimes forgot to collect the fees) and also were able to provide some assistance for the start-ups.

> *"Overall the hub works fine for us and right now [we] cannot really see what we could gain from moving to an office of our own."* (Interview 3)

However, not everyone was convinced of the current functioning of the hubs and saw them as providing the start-ups a certain type of false security, which constrained them to really see whether their applications and businesses were sustainable.

> *"I believe in accelerator hubs but I don't believe in co-working spaces. My biggest issue with hubs is that they need to implement a model that pushes you to grow because if you're going to keep me in your incubation space for three, four, five years, you are not really helping me. […] What I'm trying to say is that if we were in an incubator, we would not be where we are now as a company. So it provides safety but maybe the wrong kind.*

> *I sometimes use the comparison of a bud, an incubator is literally like a nest, after some point they push you out and you figure out how to fly or crash and die, and I think every entrepreneur needs that, if you cannot do that you have no business in running a business."* (Interview 24)

Also there were views that the focus on technology within these hubs was in general a wrong approach, and that the hubs should place more emphasis in making sure that the companies located in those were actually solving real problems, and provide guidance to the companies in this respect.

> *"If the code is not solving a problem that has been investigated, it is just another piece of code. Now if you turn that around, for example a guy who has worked for [our company] for three four years, went into these labs, and engage three or four students to work on a problem that has already been researched, […] the students were no longer writing code out of excitement or producing code, but they would be producing code to solve a problem that has been well investigated."* (Interview 18)

5.2.4.2 Constraints in Building the Applications

During the developing of the applications, certain additional constraints were also discovered, which in some cases were related to issues like lack of legislation.

> *"We're building a lot of stuff that is, for the lack of a better word, ahead of time, at least for us here, and that you can see also regarding the clinical testing which lacks behind partly because of the policy, because there is a policy for drugs but nothing for devices."* (Interview 24)

Furthermore, although talent as such was not an issue, it was sometimes difficult to find people for specific roles and also the size of the start-ups sometimes restricted what they were able to build during the development of the application.

> *"I think it's hard everywhere to find a tech lead."* (Interview 10)

> *"We had a small engineering team so we had to prioritise, we couldn't hire more people because we weren't making that much money."* (Interview 21)

Also as discussed earlier, the complexity of running a business displayed itself by requiring skills that the developers had not initially given lot of thought.

> *"Here we are mainly electronic engineers and business analysts. That really helped with the whole team dynamics, because we could develop things as engineers, but we needed someone who was more astute regarding business, to push us into the right direction."* (Interview 12)

Sometimes the difficulties eventually meant that the start-ups gave up on developing the applications.

> *"Another one that we have been working on is this Uber-like service, we actually launched half a year ago but ran out of money."* (Interview 24)

5.2.4.3 Increased Technological Resources

The development process of the applications appeared from a technological perspective to be relatively straightforward, although many of the estimations on how long it would take to develop the applications often turned out too optimistic. As noted before, once the technological basis for the applications was decided, the actual building of the application was often relatively uneventful and mainly involved the developers working on the programming. In case there were functionalities or challenges in the applications the developers did not quite know how to build or resolve, the usual approach was to search the answer from the Internet, or in the hubs turn to colleagues and ask for help. The two hubs that the researcher spent most of the time did seem to have persons that although not officially pointed to the job, were often used as points of contact in case obstacles regarding programming were encountered (Fieldnotes 4, 8)

What was noteworthy is that the situation regarding some of the initial technological constraints was not static, but over the period of data collection phase these obstacles were sometimes removed, at least to a certain extent. One of the clearest examples of this was mobile money integration to many of the applications, which was originally seen as not possible or at the very least problematic (Interviews 3 and 10). Also the release documents showed that some of the applications became more complex as well, and had added additional functions like maps to their applications. It was not clear how popular these functionalities were, but overall did point that some of the obstacles became less prevalent over time.

Other obstacles did remain though, and especially when reaching out to the poorer segments of the society for example the lack of smartphones continued being an issue. This merely points to the fact that the spread of technological capabilities was not even as such, as some areas remained problematic while others were solved.

5.2.4.4 Language

An interesting detail regarding the application development was the choice of language. In practically all the cases English was the language of choice. This was not surprising as such as English is one of the official languages of Uganda, but it was more how local languages were not usually even considered.

> *"English was an obvious choice and it doesn't really matter that much because here you just buy and sell, I mean some of the users might use local languages in their posts or like a mix of languages, but the app works in English. But we do have to think about the localisation of content, but if you do that too much then you can't go cross-border from one country to another, so like the user interface, there's no point in localising that."* (Interview 31)

One of the reasons for excluding local languages was the large variety of them. Also the developers shared the assumption that people in general spoke English, and there was no need for any other languages. Furthermore, if languages were to be added, those were usually seen as being languages like French or Spanish, and the reason given for that was often that it might make sense in relation to scaling up.

> *"English, that's the way it's going to be right now and one of the reasons is also that instead of writing reports you can just use this to share data with others. So that part is going to be tricky if it's in French for example and the next one has to change it in Portuguese […] that [local languages like Lugandan or Swahili] won't be needed because the people who work in these will in general speak English."* (Interview 14)

Whether these estimations on the need to add other languages were correct is difficult to evaluate, and for example how widely English was spoken outside Kampala was not entirely clear. The choosing of language could be also connected to certain demographic and socio-economic factors, so for example if the application was designed for a smartphone, the likelihood of a person being capable of having a smartphone and not speaking English was not generally seen as very high. Using English in the applications, especially in the mobile applications, was possibly also connected to the ways the start-ups liked to be seen and connected to the other start-ups elsewhere in the world, as they often made references to other globally known start-ups or talked about start-ups and applications more on a global level (Fieldnotes 3 and 7).

5.2.5 Launch of the Applications

Being able to launch the applications could already be considered a relative success, because there were a few start-ups that never actually reached that stage, despite having been able to build and develop the applications already quite a bit. There were also cases where the applications and sometimes even the start-ups themselves 'disappeared', which seemed to point that they had given up on developing the applications. Sometimes the focus of the start-ups also changed during the development, and as a result, the application, despite efforts to develop it, never got officially launched (Fieldnotes 6).

5.2.5.1 Delaying

In many cases the start-ups had to postpone the launching dates of the applications, for example due to financial reasons.

> *"And if we had enough money, like right now the pH sensor we want to test we need to import, and [we are] planning to break it down and see if we can improve our own version and to create a whole new product, but getting the money is difficult. Just to get one sensor for a try out is 600000 shillings, just one. And in order to do this properly you*

> *need at least 10 pieces, in order to get accurate results. So that is why the project is also going slowly, and everybody is taking the money from their own pockets, cause like me, I am paying my own tuition, which is about 1.8 million Ugandan shillings, and you have to pay your rent, your food and get some pocket money."* (Interview 23)

As noted above, it sometimes occurred that the development of the application took somewhat longer than expected due to technological reasons, but also the other, non-technical factors, which conditioned the applications' ability to function as expected, were an issue.

> *"We're just gonna delay the launch because we need to build the community first. I mean of course we have targets in terms of time but it's difficult to say we will launch at this date because we wanted to grow and mature, I mean you could already launch but I would like to have the target numbers of users first, like to grow the community to a certain level before."* (Interview 31)

Despite the delays, many of the start-ups did eventually manage to launch something, or at least reach a testing phase where the applications were tried and tested by the targeted users.

5.2.5.2 Usage

The usage of the applications tended to start quite slowly, and in some cases where the application formed part of a service that was provided for users, it also happened that the service overall was used but the application somewhat less so.

> *"Another thing is that we don't have a lot of app usage, I mean people like the service and all but they are not using the app, so it is like the app is right now the main thing. We are hoping that once we get the new version out that will increase the app usage […] But for us the main question is why people are not using the app, that is something we need to find out, we have like less than 1000 downloads, and we want it to be around 100000, so there is quite a bit of work in that."* (Interview 10)

The reasons for the slow adoption were linked back to cultural factors or just to the general quality of the applications.

> *"People like their traditional ways of doing this, they do not want to change. Overall, Ugandans are not early adopters, they start using things very late, so it is difficult to get people to use these applications, I mean, there are some, but not very many, and people are suspicious of using the applications."* (Interview 8)

> *"I don't know one single person that has installed Ugandan apps to their phone, they are often just copies from something else, and do not work that well."* (Interview 54)

As seen above, the general perception that people had of local application development was also an issue, as sometimes applications were just seen as something that young people built but were not to be taken seriously (Interviews 59 and 63). These kinds of perceptions mattered as one interviewee pointed out in relation to dealing with partners or building applications for local companies.

> *"However, if I am in an incubator and I don't have offices, I just have a desk, then asking 10 million, even I would be like what, what am I paying you for. Especially here in Uganda people do want to come to your premises, that is what shows authenticity, another thing is we are actually buying a company car, and that's for business, not to move around, so it's more about image. Sometimes when I'm about to close a deal and then I jump to a boda boda [motorcycle taxi], and you see the look on the guy's face, like we just agreed a 20 million deal and this young man probably just wants to get rich from me, whereas if I do this with a car, no matter how small the car is, the client won't think twice about it."* (Interview 24)

Some applications and start-ups did however manage over time to attract bigger crowds, although in general the success rate of start-ups reaching large audiences was not very high. In order to gain attraction, the key factors that were mentioned where often around being able to solve a problem that existed in the society and not only in the developers' minds. Marketing was also another issue, as many companies could not really afford big campaigns.

5.2.5.3 Marketing

In general the start-ups did not have the resources to market their applications on a large scale, although social media channels like Twitter and Facebook was used widely as it was claimed that they provided a relatively inexpensive way to reach bigger audiences. However, this type of marketing did not necessarily reach the targeted users, and in these cases a typical way to market was simply to go and meet them and hope the word would be spread around.

> *"Yeah we had a problem with marketing cause you know, you have to explain this to someone, it's not like selling hardware like a computer where they can see the product and touch it. Doing service especially in the tech business is still something quite new so you have to convince [the customers] but now we're past that because people know what we're doing."* (Interview 21)

> *"The thing is I knew where the markets are, cause the buyers are bit hard to get to, but their locations are also few, so it is very easy to narrow down the locations and taking the courage to go and speak to them and listen to them."* (Interview 18)

Also some of the start-ups had gotten coverage in international media, usually in stories which discussed application development in developing countries. This did not necessarily help in getting funding for example, although there were hopes that it might get more people to use the service.

> *"The media attention we have gotten has been ridiculous, we've been covered by Le Monde, BBC twice, CNN, [and this] because it's an interesting idea, which is good as long as you don't take it too seriously".* (Interview 10)

> *"We have gotten attention, like the BBC came here to see what we do, we went with them to meet the customers, and publicity is good, might help in relation to getting people to sign up."* (Interview 3)

> *"It [publicity] does not really help with people wanting to invest to the company, there has not been anything pointing to that direction."* (Interview 4)

In the rare occasion that mass marketing such as radio adverts had been done, the results were not entirely clear.

> *"I think the buyers also are looking for easier routes to sellers, so indeed, it was more footwork to talk to the buyers, but the good thing with them is that you don't need that many of them, and the other bit is also radio advertising, so again with a limited budget we had once a week a radio ad, so that brought in a couple of buyers but they were not sticky buyers. Those are buyers who are more browsing, they want to see if the platform is really working, but that gave us an opportunity to talk to them, cause once they connect to the platform their numbers are captured and we can call them. But I cannot say we have had too much success there."* (Interview 18)

In general, it appeared that the usage was largely driven by word of mouth and by the start-ups talking directly to the customers, as it was unclear how successful marketing campaigns overall were. Talking to the users was also helpful in terms of getting feedback and incorporating that into the applications.

> *"I just want to keep on getting feedback and developing this so that it matches well the requirements of the schools. And I want to hire more people because just giving support and developing this especially with a lot of schools might be too much alone. Overall it would be great if I get the parents involved because they can then recommend this program or software to the schools saying 'this is great'."* (Interview 19)

5.2.6 User Feedback

> *"We also got feedback regarding the name, one of our developers wanted to call it VaginaStick, but people were like 'oh that name is so scary, it looks like such a scientific term', so the negative feedback was mainly regarding the name."* (Interview 23)

As the start-ups progressed with their applications, it became clearer that they needed more feedback from the users to understand the issues that hindered the success of the applications. There was a division between start-ups regarding the moment they approached the users. Many of the more successful applications seemed to do this in the very beginning, before even starting to build the applications, while others built the applications first and only then reached the users to understand the issues better. However, it was widely acknowledged that understanding the problems thoroughly was a key in creating successful applications.

> *"So we think we cannot succeed unless the user is also involved. So in practice you just go out to the clients, and check if they like your product or if they criticise it, you use the feedback and we tried to take that feedback and put it into something that they can really use."* (Interview 31)

5.2.6.1 Relation to Users

Getting feedback meant establishing relations with the targeted groups, and in order to do this it was in some cases necessary to first have at least a working prototype of the application. However, the application itself did not yet guarantee that users were willing to try it, which further highlighted the non-technological aspects of running a start-up.

> *"The company that wants the software, it might be that people still do not use it in the company, and then I cannot get feedback on what to change in the program, so other than technological things give a lot of work."* (Interview 8)

Also depending on the type of applications, cultural norms and attitudes also made it sometimes difficult to collect user feedback.

> *"Yes it was part of the Imagine Challenge Cup to go and ask people what they think, and we got good feedback, like if I can test myself in private and I don't have to go through the trauma of opening my legs in front of the doctor […] Although the problem was also that since these things are private, people don't want to talk about it too much."* (Interview 23)

Many also highlighted the need to be actively in contact with the users, and that this relationship had to be ongoing.

> *"Usually we tried to go to the communities and so on, instead of meeting the farmers individually, unless they're doing something really new that excites us, like maybe using new methods, planting using new seeds, new fertilizers, and we want to see that and understand the story, and we can also take out other farmers there to see."* (Interview 21)

Another reason for the ongoing user interaction was the changes that occurred not only within the targeted user groups themselves, but also within the market. Those marked the shifting attitudes among the people, and with that, their overall perception of the problems that had to be addressed.

> *"There is a market, we have a fairly large expatriate community, we have a growing middle class, and with that I mean that they are also starting to appreciate the so-called finer things in life, like services, like why should I sit on a boda [motorcycle taxi] and possibly die when I can sit in one that has a helmet, and I protect myself somehow. So you are getting these young, we call them young corporates here, around 25 [of age], [who] have a first job, have their first car now, that sort of things."* (Interview 24)

Once the relationship was established with the users, the next obvious step was to get the feedback from these users and incorporate it into the applications.

5.2.6.2 Incorporating Feedback

Being in contact with the users often led the developers to realise that they had not taken some important factors into account, had added functionalities that were not needed or that they had not fully understood the problem they were trying to solve.

> *"We also removed stuff like the market prices because we got feedback that there was no need for that."* (Interview 31)

> *"Like take the component of roads and distances, that alone was complex because we don't have Google Maps to show users how this works. We went through various options to see how we can represent this on SMS, like how the seller could send a code that could indicate the type of road surface to their location, and all that struggle came down to a*

design that was so simple, cause again we had to go back and interact with the buyers and sellers and to check what they really care about. And it turned out that as much as that is important, but it is already an expectation what roads in Uganda and in developing countries will be, so we were worrying about a problem that was not a problem." (Interview 18)

This type of reinterpretation of the problem occurred to some extent, and also in some cases the problem that the start-up had identified and tried to solve was not actually related to the group they wanted to target at first, but was more of an issue for another group.

"I mean the muzungus [white people, tourists] are the ones that usually ask for the helmet, but the locals do not, so overall the local habits are a bit of a problem, people just do not want to use helmets, and I don't know, maybe the concept of safety is a bit different, like everyone knows the traffic is dangerous, but the helmet for example, they just do not do that." (Interview 10)

The interaction with the users could also help the start-ups to identify another problem area that they had not thought of.

"So in the beginning we were doing tracking, like car tracking, and as we were doing that our client started telling us that 'then we have problems with the Ugandan mechanics who are always robbing us, that is there a way that you can develop an application, which can keep track of all the things that have been done and notify us'." (Interview 12)

In sum, lot of the feedback that the start-ups got was not so much about the technical aspects of the application, but more about understanding the context and the user groups that the application tried to reach.

"There is such a thing as developing an application, but we also do this thing called empathy map, and that is something that sets us apart from the other start-ups, the thing is that we need to be fully absorbed in the emotions of both the participants, that means the person who has the problem and also the mechanic. So we spend a day with actual mechanics, we sit there, we talk, even about the ways how they cheat a client, and thanks for this it is easier for us to sit down here and start coding and come up with something that will really help them." (Interview 11)

5.2.6.3 Scaling Up and Updates

"The plan is to go worldwide, and we already have a person looking at that". (Interview 23)

Scaling up to new market areas was something that basically all the start-ups had been thinking about, but very few of them had done so or were even at a stage to start taking steps towards it. In most cases the areas nearby were the most common choice, such as other East African countries, which was partly not only due to their geographical proximity but also they were seen as places where the societies shared similar issues. Also lack of financial resources was sometimes mentioned as reason to choose particular countries.

> *"I rather go to Guinea Bissau where with half the money you can get so much more, [much] wider reach than for example in the UK."* (Interview 20)

Many also acknowledged that the inability to automate some of the functions at that particular time made it near impossible to reach other market areas. As noted earlier, this meant more manual work for each customer, and if the numbers of customers were to grow, it would have been very difficult for the start-ups to keep up.

> *"The most challenging part has been the technology, I mean the whole divide between smartphones and feature phones, and because of that we end up being not a platform and that is what we want to be. I mean ideally we don't want to get rid of the whole thing completely, we still want to do with some of that support thing, and have people online all the time to be able to support and ready to help if something goes wrong, but that model is not scalable. If we have 100000 people, there's no way we can make 100000 journeys every time, it is just so much more work. So the smartphone takes this kind of work away, and enables us to be what we want to be, a service that works through the phone. […] but if you cannot connect these things with the technology, that just means so much more work to us, which means that instead of scaling the platform we have to scale up the team [in order to] to service all the demand and then that's also when the costs go up."* (Interview 3)

As a whole, encounters with users enabled the start-ups to not only learn about their targeted user groups but also understand better the local nuances that affected how the targeted user groups viewed the world. As a result, they gained a more accurate picture of the needs and wants that these groups had or the problems that had to be solved. This again opened the way for changes both within the applications as well as in the overall functioning of the start-ups, although it was not given that these changes were eventually done.

5.2.7 Adaptations and Updates to the Applications and Business Models

Out of all the start-ups that were studied for this research, there were some who showed great promise and even those that one might call as successful. However, most of the start-ups appeared not to get that far, at least not instantly. Furthermore, many of the start-ups had to do sometimes even rather drastic adaptions during the process both to the applications and business models, or alternatively shut down the start-up.

> *"The start-ups don't change much the idea, and they build things for themselves but not necessarily thinking the users, and some of those are relevant and well, some less relevant."* (Interview 40)

Sometimes the adaptations that were done as a result of the user feedback were either relatively small and could be taken care of with minor changes to the applications or other non-technological areas. However, in some cases there was a need to do bigger changes, which often took the form of one of the three forms discussed below.

5.2.7.1 Change of Technological Basis

Changing the technological basis did occur occasionally, although overall this was relatively rare. If the technological basis had been chosen wrong and the application could not therefore gain users, it was more likely that the start-up simply withered away. In some cases however, there was indication that if nothing else, the start-ups tried to add functions that for example made it possible for the users to use basic phones to use the service instead of having a smartphone.

> *"We don't leave the other operating systems out because we also have a number you can call."* (Interview 3)

In other words, sometimes functionalities had to be added to the applications that enabled the usage of the service with less developed technologies like basic phones. The lack of certain technological resources or capabilities among the targeted users also led the applications to be changed so that they would work in these locations. This also meant that instead of removing some technologies, a new level of complexity had to be added.

> *"The schools wanted an offline version, because of the Internet connection, which is not that good [… and] teachers don't have laptops so they need to go to the lab, so they told me it's a good idea but it only works online […] Let's say I have to create reports and then I have to push them up to the system later and if you have no Internet connection there's not much you can do. But all this gives me just a lot more work […] […] Here you can see an overview of the program, and this is offline and whenever you go online it syncs automatically."* (Interview 19)

Lack of technological resources could also refer to the type of data that was available, and as a result the developers had to find alternative sources for data and to replace the more common ones used for example in the Western countries such as social media.

> *"For example the radio can be a useful tool here, as there are a lot of small radio stations that are community run for example. We can use those to see national disasters, disease outbreaks and so on, but this requires speech recognition to automate the data collection."* (Interview 46)

Changing the technological basis meant additional work, and this change was not always about making the application less technical and turning the provided service more manual, although this was more often the case. At the same time, some start-ups refused to change the technological basis and instead saw this more as a matter of time automatically correcting the situation in terms of lacking technological resources as more people would acquire smartphones and so forth. In addition, it was not always clear how necessary applications overall were.

> *"Like three years ago everyone was building these apps and everyone was saying how these are going to be big hits, and guess what, three years down the road, who is using the apps. And there was so many. […] There was this one [idea of] providing smartphones for farmers and [to get] those guys using them, change their mindset. [But] they might just use it for mobile money and [they] say we don't need any of the other*

> *stuff. And this whole USSD thing, maybe the numbers are good but in the end they probably even end up costing more in terms of sending SMSs and everything."* (Interview 12)

As noted, the technological basis was not very easy to change, and it also often came with additional costs for example in terms of increased manual work in running the business. Overall, when faced with situations where the chosen technological basis did not match with the targeted users, another strategy was to target other user groups that matched better the chosen technologies such as smartphones.

5.2.7.2 Targeting New User Groups

If new user groups could be identified that did not require change of the underlying technology, it was somewhat easier to change the targeted user group instead of changing the technological basis. Overall, the reasons for changing targeted users were not only limited to technological ones, but sometimes the targeted user group was not the best possible one for example due to motivational reasons.

> *"I mean for them there is little incentive to use the app, for the monitors, who see little benefit in it, although for the rest it is good because it is easier to send data and you don't need to transfer that into a file manually. So we are now targeting the farmers also, because they can benefit from it more, we can help them to take care of the disease."* (Interview 16)

Similarly, sometimes there was a shift or an expansion from a customer based approach (B2C) to one where start-ups started serving other companies (B2B).

> *"Some organizations are also now using our drivers for transport, which is a good thing, cause then we can try to get revenue straight from them."* (Interview 10)

Related to this, in some cases it was simply more beneficial to target entities that were more likely to pay for the services, which often meant not only people with more income, but also organisations that were willing to pay for the application or the type of services the start-ups were offering. Another factor was that sometimes it was seen as easier to try to reach customers abroad, although there was not much evidence of this actually happening, and some also pointed out to the difficulties in competing with foreign entities overall.

> *"So enterprises have always been making money from way back, and sometimes it makes sense to go abroad because you can charge higher fees, and there is also this issue of local trust, people don't trust the local developers […] And this is also where the enterprises come into play, they have been building managements systems for schools, hospitals, supermarkets, and all this locally. Enterprises have a client before they build the product, so locally they are making money, but most of the money goes to the foreign funds, because most of them are government deals and very huge infrastructure projects, that actually most of our companies are not in a position to get."* (Interview 44)

The comment above also points out the importance of tapping into existing flows of money and generating income, and targeting other companies were often seen as a guarantee of this, especially if compared to the poorer segments of the society. Targeting the poor was also more difficult due to the other factors mentioned above, like lack of access to smartphones. Overall serving the poor and making revenue out of that was considered quite difficult, yet there was an area where they were more often targeted by the applications and start-ups. This occurred in platform-based solutions, where the poor became providers of some particular service to users who were willing to pay for it. In these cases the applications functioned as platforms, i.e. bringing the two groups of people together. However, as can be seen from some of the previous findings, even then the lack of technology among these groups was sometimes problematic for the start-ups.

5.2.7.3 Application as a Side Product

When faced with technological constraints, the third strategy was to move the focus away from the application to the part of the business that seemed more promising. In most cases the applications formed part of the service start-ups provided, but not the whole service. For example an application linking motorcycle taxis to customers, the application did the linking, but in addition there were other areas of the service such as taking the customer to the wanted location that naturally the application could not do[11]. In other words, the applications were designed to mediate or to process a particular aspect of these events that took place in the particular location. What in some cases occurred was that the needs or wants that the start-ups tried to address were valid as such, yet the application itself was not used. However, there was interest among the targeted users towards the other aspects of the service the start-up was offering. That often meant that the importance of the application, if possible, was reduced in importance in favour of these other relevant aspects.

> *"There is this challenge, and I think we will have it, and that is about what is the value that we are adding [with the application] and we need to do more. […] So we're partnering with powder detergent as we're trying to get our own brand, so when you go to the supermarket you can buy our brand, so we do want to provide the service but we also want your laundry to smell in a specific way."* (Interview 18)

In the comment above, the application was no longer related to the new area of business, which was to produce the start-up's own detergent brand. Therefore, something that had started as a pure tech start-up and based on the functioning of an application, moved away from this as they looked to expand their business and in a certain sense, became more of a regular start-up

[11] Exceptions of these were for example most video games, where the games could often be played in a device without the need for any connection to other, non-technological areas.

instead of a technology start-up. Similarly, sometimes running a start-up gave additional ideas on areas where to do business, which had little to do with an application or even with anything technological.

> *"And another thing that I have been working are t-shirts, the market is huge, so in here you can get good material and price, and we are selling everything offline now, and we are raising some money. There is a lot of promise in other areas that are not about tech, and that is being overlooked cause people have not really understood the opportunities that are in the market, and you can do a lot of things without tech."* (Interview 44)

5.3 Conclusion

In sum, the original ideas for the applications and businesses stemmed from the occurrences and routines in the local environment, which were further shaped by the existing cultural norms and other contextual factors. Furthermore, issues like lack of funding also impacted the original ideas of the applications, as every idea had to have some link to receiving money, be that in terms of generating revenue or receiving funding from a particular source. After having the idea, there were some intentions to verify the ideas, although this sometimes also occurred later in the application development process. In many cases it turned out that the original idea did not really match the realities of the targeted users or that certain perceptions of the problem the start-ups had identified were not really shared by the users. However, not all of these issues were problematic and occasionally the new findings regarding the original idea were actually something that the start-ups could turn into their advantage. In addition, technological constraints, be that in terms of access, technology, or lack of available resources, became more visible and also affected the validity of the idea in relation to turning the idea into an application.

In relation to building the application, access to developers was sometimes seen as a problem but in most cases the people who had gotten the idea for an application were developers themselves. Overall, lack of technological skills was rarely an issue, and if the developers did not know how to do something, there were ways to acquire that knowledge such as asking other developers or using online resources. The skillsets a developer had also affected the type of technologies that were chosen for the applications, and furthermore institutional factors, such as companies promoting their application development tools, affected these technological choices. In addition, the start-ups made types of bets on what the future situations would be like for example in relation to smartphone penetration, and built their applications accordingly.

Most of the start-ups included in the study resided in the tech hubs. Some saw that the hubs were not providing the start-ups with the right kind of support, as more help would have been needed in understanding the problem the start-ups were trying to solve, instead of learning different technologies. The actual building stage of the applications appeared often relatively

straightforward although time consuming because of certain contextual and technological obstacles. Over time some of the initial technological obstacles, such as integrating mobile payments to the applications, were removed. An interesting finding was also related to the language used in the applications. Despite the large number of local languages, English was always the first language for the applications, followed by languages like Spanish or French, and only one or two start-ups had even thought about equipping the applications with local languages. There was some indication that English was chosen also because it helped the start-ups to identify themselves with the larger community of application start-ups across the world.

Some of the start-ups never reached launching phase, or the businesses they were trying to create gradually became less dependent on the usage of the application. Many of the start-ups had to delay the launches, and when those did occur, usage was not guaranteed and it often took a lot of time for the start-ups to get sizable user bases. Interestingly enough, international media like BBC, CNN or Le Monde had made quite a few reports from some of the start-ups, but those had not led to more funding or significant increases in the numbers of users. Marketing overall was done using social media due to its low cost or then just using word of mouth to reach more and more customers.

The importance of user feedback was understood quite well, but in many cases acquiring feedback was done only after launching the first versions of the applications, whereas the more successful start-ups seemed to have done this in the very beginning. Overall, the problems that were encountered were in many cases not so much about directly linked to technology, but more about understanding the problem that the application tried to solve or then getting to the bottom of the need or want the application tried to address. At the same time, there was some indication that the markets and targeted user groups were also changing in relation to the problems they faced and needed solving. The applications and also the businesses were adapted according to the feedback, sometimes at the expense of the role of the application. The goal for every start-up was to scale up to new markets, but among other things lack of resources and the amount of manual work that the start-ups had to do because of inability to automate certain functions hindered these aspirations.

In cases where the start-ups had realised that the technology they had chosen for the application did not manage to meet the needs or wants of the targeted users, the start-ups basically used one of the three following strategies. One was to change the technological basis of the application, for example by building an application that worked in a basic phone instead of requiring a smartphone. The second strategy was to identify new user groups that had the same want, need or problem that the original user group but also made a better match regarding the

used technology. The third option was to make the application less significant in the overall business, and as result, concentrate more on the other areas of the business over the application.

The findings presented above highlight the areas where location has a role in application development. Overall, the technological resources did not appear to cause many problems for the developers in terms of having the skills and knowledge to use them. Although it has to be remembered that it was often the developers themselves doing the evaluation, it seems safe to argue that technological skills were not the main barrier for the success of the applications. However, other technology related factors, such as general access to technology across the whole population, were an issue, yet in a manner that seemed to go deeper than having the devices and the ability to use the technologies appropriately.

Overall, the findings show how application development is far from a pure technological or even business creating process, but goes well beyond these and ultimately ties down to local contextual factors. In the analysis chapter that follows, we draw on our conceptual framework in order to interpret our findings.

6 Analysis of the Findings

> *"It is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail."* (Maslow, 1966, 15)

This chapter draws on the conceptual framework presented earlier to analyse the findings presented in the previous chapter in order to answer the research questions posed at the outset. To recap, the main research question "*how does context impact the application development process in a developing country?*" is further split into three sub-questions, and these three questions are the following:

> *Q1. How do societal routines display themselves in the application development process?*
>
> *Q2. How are technological resources chosen to develop the applications and how these resources match with the developers' ideas for the applications?*
>
> *Q3. How are the applications received by the targeted users and what kinds of implications this has for the applications?*

As can be seen from these questions, they follow the logical order of application development process, starting from the ideas for the applications and then moving on to the actual development of the applications, and finally concentrating on the question regarding the reception and usage of the applications by the targeted users. The aim for this chapter is to develop a model that takes into account both the contextual factors affecting the application development process and the role played by technology, as well as the interaction between the two. This is done by the identification of particular mechanisms that could have led to the outcomes and findings described in Chapter 5.

6.1 Mechanisms

Within the critical realist ontology, the empirical is what can be observed but what constitutes the real are the underlying causal mechanisms (Klein, 2004). In the following, the intention is to present mechanisms that could lead to the particular observable events in relation to application development described in the previous chapter. The process of doing this is called retroduction, in which the idea is to *"take some unexplained phenomenon and propose hypothetical mechanisms that, if they existed, would generate or cause that which is to be explained. So, we move from experiences in the empirical domain to possible structures in the real domain"* (Mingers, 2004, 94-95). The purpose of identifying these mechanisms is to further understand the role context plays in this process and in relation to the technological components of application development.

It is difficult however to be entirely sure that the identified mechanisms are the ones that truly explain the observed events. Therefore, in order to approach the accurate understanding of existing mechanisms, analysis based on critical realism should come up with alternative mechanisms and compare between them to decide which one best explains the observed events. As a result, at the end of each section there is a discussion of the other mechanisms that could have provided similar observable outcomes and why they were rejected.

The mechanisms are organised according to the stage of application development they have most impact on. The first section discusses the role of societal routines in the ideation stage of an application while the second section analyses more the role of technology in the actual development of the application. The third and final sections pay attention to the usage of the applications and by doing so draw mostly on the concepts of code and space as discussed by Kitchin and Dodge (2011). However, similar to the previous chapters, these categorizations are not exclusive, and as a result some aspects of routines can for example also be identified in sections discussing the development of the application or their usage.

6.1.1 Routines as Basis for the Ideas of the Applications

Our conceptualisation of context through the lens of routines provides a useful starting point for analysis.

Feldman and Pentland (2003) define routines as "*repetitive, recognizable patterns of interdependent actions, carried out by multiple actors*" (p. 95), and although used widely in research of organisations, there has been less use of them on a broader level such as societies. However, just as in organisations also in societies people perform particular routines on a repetitive basis, and overall these routines are linked to other routines. The shape and form these routines take depends on a number of contextual factors, such as cultural norms, socio-economic variables and availability of technology to name a few. In relation to the critical realist notion on mechanisms, the performance of routines can be seen as forming part of the empirical. As Klein (2004) describes the concepts of real, actual and empirical in critical realism, "*the real are the causal mechanisms and structures that produce actual events a subset of which then is empirically observed*" (p. 131). Following this line of thought, the contextual factors equal to the 'causal mechanisms' and the reasons for performing routines are then the 'actual events'. Certain aspects of those events or routine performances can be observed, thus forming what in critical realism is referred to as the empirical. However, the notions of real, actual and empirical appear to have a dynamic character as the above mentioned descriptions on what falls under each of the three are different when moving on to discuss the mechanisms that impact the ideas for the applications. In this case, the so-called empirical, i.e. the part that can be observed, are

the applications themselves, and the actual events are formed of the routines that form the basis for the applications ideas. The real on its part consists then of the affordances made available by the technological resources in addition to the contextual factors.

This is noteworthy since within the confines of this chapter, the intention is to reveal the mechanisms affecting the development of the applications but not so much the mechanisms that led to the emergence of the routines that affect application development process. At the same time it is important to be aware of this distinction between the two and notice the certain level of relativity in them. Routines and their performance in relation to context are seen as the empirical and are largely observable, but within application development, they become part of the actual events that are less observable for the researcher and to the application developers themselves as can be seen from the analysis below. Overall, the findings and the analysis show the applicability of the usage of routines in understanding the role of context in application development[12].

6.1.1.1 Perceptions of Existing Routines

As pointed out in Chapter 5 regarding the origins of ideas for applications, the basis for the ideas behind applications was a perception of a particular event, which the start-ups' founders had perceived in the local context. Examples of these were for example fire brigades not knowing where to go (Interview 28), students of less good schools not having access to adequate study materials (Interview 19) or high numbers of people suffering from cervical cancer (Interview 23). These events were caused by more complex contextual factors, such as lack of proper system of addresses or not having the money to access good study materials. It was difficult to say with certainty which were the exact reasons that had led to the people to observe particular events and then use those as a basis for the idea of an application. However, often the events that were chosen were the kind that people behind the start-ups had experienced either directly themselves, or through relatives and friends. The point here is that those events were not chosen at random, but some sort of connection, be that through personal experiences, work or friends, existed between the event and the ideas for the applications.

[12] However, it is worthy to mention that there were one or two cases where the newly constructed and technologically mediated routine did not necessarily match entirely the definition provided by Feldman and Pentland (2003) due to the changes that occurred to them in the application development process. This was the case regarding games, where something that was to a large extent a societal routine lost some of the routine characteristics when it was turned into an application. An example of this were games like card games, in which a routine that in many ways in its existing form consisted of interlinked actions and multiple actors, became something that was performed usually alone and consisted of one, somewhat isolated task. As noted, this applied largely only to games though, whereas other type of applications maintained their routine type characteristics as mentioned in the definition even after they underwent change in the application development process.

Although events can be isolated and possibly occur only once, it was more common for the events that were behind the ideas that they kept on repeating on a regular basis (Interview 10). They could be seen more as statements on how the contextual factors became visible in the society, and when observed, how the observers understood these events and their overall functioning. These events often simply presented things how they were, without labelling them in any particular manner such as negative or positive, efficient or inefficient. For the start-ups, it made more sense to view the events in the form of routines and routine performances. In practice, this meant that instead of the relatively static notion of events, they were transformed into routines and into the performance of those routines. As a result, '*fire brigades not knowing where to go*' became a routine of '*fire brigade organisations (or some other entity) taking more time to arrive because it was difficult to find the locations*', or '*student from poorer schools who did not have access to good study materials*' took the form of '*better schools providing their students good material*'. In addition, part of the performance of the routines could be understood as non-action, such as people not going to check if they had an infection that may lead to cervical cancer (Interview 23). Figure 6.1 shows the process of how societal routines formed the basis of the ideas for applications.

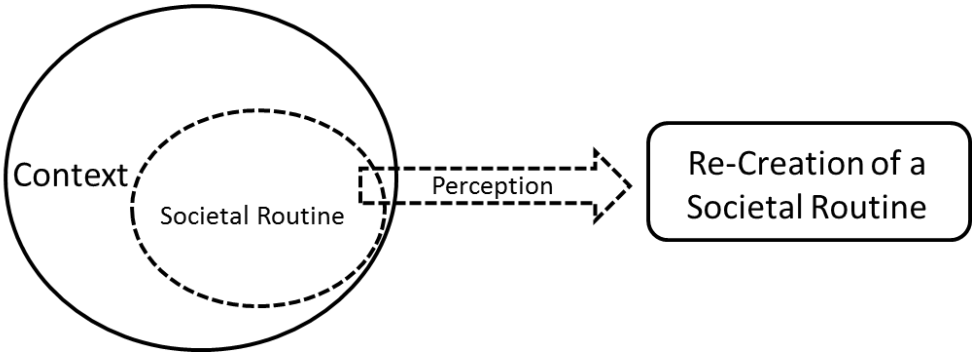

Figure 6.1. Events as Basis for Perception and Recreation of Routines

The key point for transforming the events into routine performances was that it enabled the observations made by the start-ups to be interpreted as sequences of actions instead of purely static occurrences. This is in line with the definition of routines by Feldman and Pentland (2003), which mentions how routines consist of *'interdependent actions*', and therefore a routine can be described as a series of stages that each play a particular role in performing the routine. This partition of a routine into particular stages enabled the developers and start-ups to see where and at which stage the existing routine performance could be improved and how the application would fit into it. In order to do this, the different stages of the existing routine performance were given characterisations, which in most cases could be seen as value judgements on how well the particular routine stage functioned. Regarding particular stages these characterisations pointed out why the routine needed changing, what was the problem within the routine that

needed solving, or what kinds of needs and wants the existing routine performance did not address.

6.1.1.2 Attaching Attributes to Perceived Routines

This process of giving characterisations to the stages of routines could best be described in the form of giving routines certain value-based attributes such as inefficient, slow, or costly. As can be seen, the attribute was usually negative as it was indicating a flaw in the routine performance, and the point for the application was therefore to enable these negative attributes to be converted into more positive ones. However, similar to the original perceptions discussed above, there was no guarantee as such that the view of this flaw or problem in routine performance was shared by the targeted users.

Interviewees mentioned how some stage of a particular societal routine was for example unsafe (Interview 10, using motorcycle taxis), disappointing (Interview 7, lending money to relatives), or left people poor (Interview 3, washers doing laundry). This is therefore where the value estimations came to the fore, as these attributes highlighted the perception the start-ups had about where within a particular societal routine there was a problem that needed solving or a need or want that should be addressed. By giving attributes to the stages of the chosen societal routines, the start-ups also started implying how the societal routine could be made better in some aspect and as a result, replace the older routine with the better functioning new routine.

As the start-ups were technology start-ups, this in practice meant that the solution for the problem or addressing a need or want took a technological form, i.e. at least part of the new routine performance, which intended to solve an identified problem, was now performed by using an application. However, how much of the routine performance was transformed into an application varied considerably. For example in relation to games, the routine of playing of a traditional card game could be transferred entirely into an application (if taking into account purely the playing of the game and not for example the social dimensions of playing the game) (Fieldnotes 4). Alternatively, if the task was for example to create a platform to connect washers and people needing to do their laundry, certain stages of the routine performance still remained outside the application, such as washing the laundry. Despite this kind of variance across applications, in all of them the actual solution to the problem that was identified came in the form of a technology. The stages that were given this technological form went through the process of functional simplification (Kallinikos, Hasselbladh, & Marton, 2013), where only the aspects of those particular stages that were compatible with the technology were incorporated into the application while the other aspects and stages remained outside. Using the same example of washing laundry, an application that connected the washers with the customers

provided a tool for contact, but at the same time left out areas such as if the washer and the customer knew each other already or if the washer had in the past given a discount on doing laundry for the customer (Fieldnotes 6).

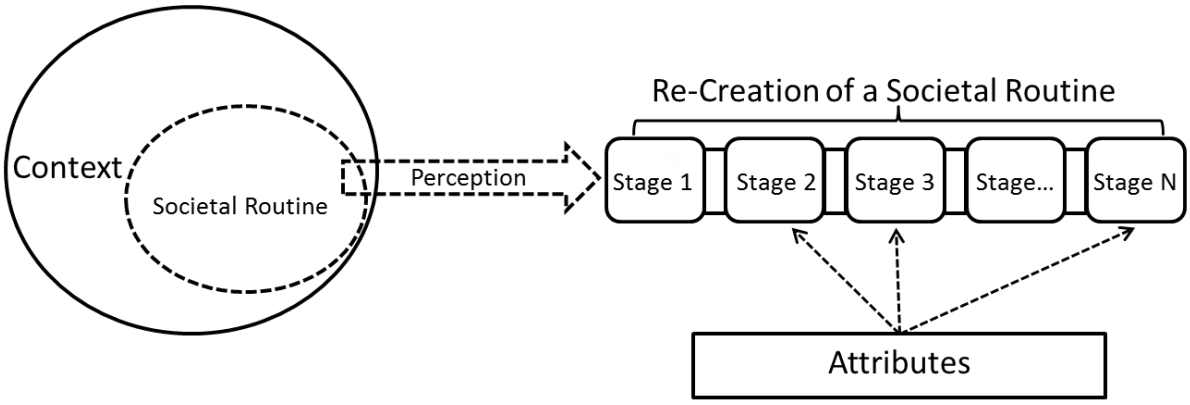

Figure 6.2. Attaching Attributes to the Different Stages of the Perceived Routine

Figure 6.2 functions as a continuum for figure 6.1 and displays how attributes were linked to the recreated societal routines. Overall, giving attributes to the identified routines was also where the possible limitations of technology first started to appear, especially in terms of what could be built by using particular technologies. Not everything could be built, and the findings showed that there were clearly limits to the affordances that any technology can provide (e.g. Interview 3). In addition, the usage of technologies also affected the ideas that the start-ups had for their applications, since the solutions to the identified problems were to some extent dictated by the affordances of technologies and the narratives around them. These narratives took the form of known technology success stories of certain well-known companies like Uber, Facebook, WhatsApp and Google. When discussing about their applications with the interviewees, they often made references to these companies, and the start-ups often made comparisons between themselves and for example Uber (e.g. Interviews 4, 10 and 20).

Therefore, the issue on how a problem was to be solved by using technology was also subject to ideas about how these technologies worked elsewhere. Whereas the societal routine was very much based on the local context, the technology in itself was mostly external, and so were the understandings about how it could be used. Furthermore, it could not be taken as given that the perception of a societal routine that the start-ups had was even correct. Often it appeared that it actually differed quite a bit from the ways the targeted users viewed the routine and its performance.

6.1.1.3 The Accuracy of Routine Perception

As the start-ups and the persons behind the ideas lived in the region, the observations on routine performances were based mostly on the local routines. What is noteworthy is that the mere fact

of the start-ups residing in the same area where the routine existed did not guarantee that the perception of the routine was accurate. As pointed out in the findings on the misunderstandings or verifications regarding the routines, quite often it turned out that the start-ups had not been able to fully understand the routines and their performance (e.g. Interview 8 and 18). As can be seen from figure 6.1 and 6.2, the observed routines were not purely taken from the local environment but also constructed in the process by the start-ups and were therefore subjective. As a result, the accuracy of these constructed routines to some extent rested on the assumption that the start-ups had identified and captured the routines in a similar manner to the way they were seen by the targeted users.

Overall, it cannot be argued that there exists an objective version of the routine that matches the reality completely accurately. Just like the start-ups, the targeted users had subjective and even in some cases contrasting views of how and why the routines were performed (e.g. 50, 51 and 52). Despite this, the gaps between the routines constructed by the start-ups and the views that the targeted user groups had of the routine were sometimes quite a lot bigger than the views between the different targeted users. As a result, there were differences between applications on how close their routine perceptions and constructions came to the views held by the targeted users.

To a great extent, the importance of understanding the routine seemed to be relatively clear to many start-ups as it highlighted the need to understand the problem first or stressed the importance of interacting with the users (e.g. interview 18, 8 and 24). Furthermore, most of the more successful start-ups had spent considerable time studying the market and the problem they wanted to solve before building any application (Fieldnotes 9 and 5). Interestingly this was especially the case (but not limited to) when the person who came up with the idea had a non-technological background, and instead of having a degree in computer science, had studied for example economics. The point here is that although there were clear cases where the routine was simply not understood correctly, the findings seemed to imply that one of the forces that led the start-ups routine construction to deviate from the views of the targeted users could be partly linked back to technological factors.

The reasons for this could be twofold. The first one relates to the role of technology, as more than looking for problems to solve, the start-ups and developers were looking for technological solutions they could provide. This view could also be backed by the comments on how they realised how much additional work had to be done not only in terms of running the start-ups (e.g. Interview 10) but also in redefining the applications and the problems they were trying to

solve (e.g. Interview 18), and especially when both the users and the start-ups seemed to agree on the problem as such but not on the technological way of solving it (e.g. Interview 16).

The second reason for the technology's role in getting the routines wrong is linked to the first ones. Although the problems and with them the ideas for the applications stemmed from the local context, the technology-based solutions no longer did. It is at this stage, when technological narratives represented for example by international companies came into play, and the solutions were in some cases imitating some aspects of famous applications like WhatsApp or Uber (e.g. Interview 3 and 4). This was also something that was mentioned as reasons why many of the local applications failed (e.g. Interviews 24 and 44).

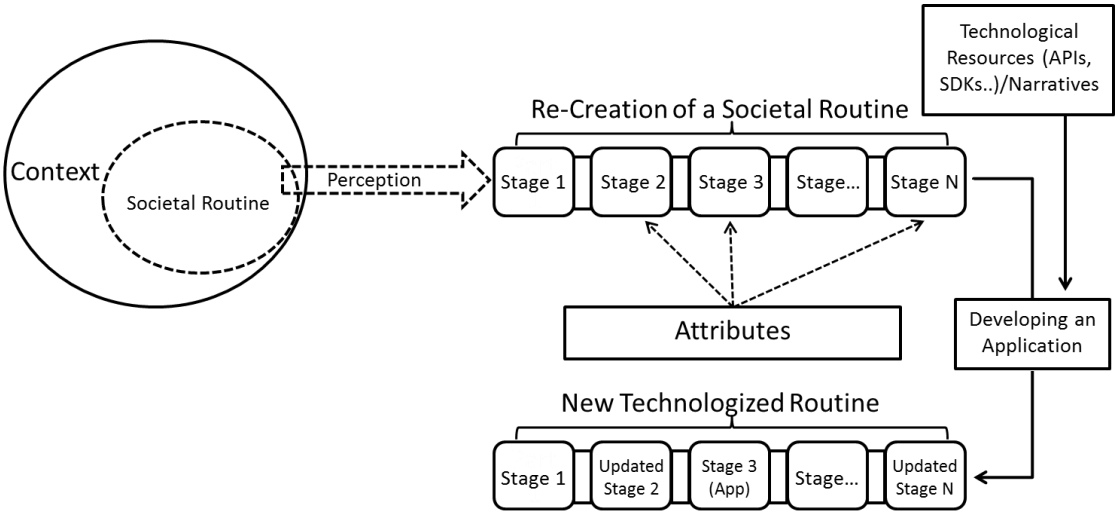

Figure 6.3. From Routine Perception to a New Technologized Routine

Figure 6.3 shows the entire transformation of a societal routine to an application and to a newly technologized routine. In sum, already the initial perceptions of the existing societal routines were often not entirely accurate, because the developers had misunderstood or misanalysed them and their re-constructed versions of the routine failed to grasp factors that were important for solving a particular problem or addressing existing needs and wants. On top of this, technology, both in its material form and also in the set of narratives that came with it, led the newly constructed and technologized routine to deviate even further. However, what is clear is that the basis for the applications was in almost all the cases a particular local routine and with that, the local context. This alone though did not guarantee that the routine that was then transferred into an application was necessarily captured accurately in a manner that helped to solve real existing problems or to address people's needs and wants.

Overall, if the new technologized routines were accepted by the target groups and incorporated into the local context as new ways of performing particular routines, they would have also

changed the underlying context. From a technological perspective one example of a technology that had clearly impacted the local context was the usage of mobile money. Several of the interviewees mentioned this and furthermore wanted to build their applications in a manner that enabled the usage of mobile money (e.g. Interviews 4, 7 and 10). The point here is that the context itself is dynamic, and keeps on changing and adapting according to the newly acquired routines, just as well as to changes in attitudes, beliefs and norms. Therefore also temporal factors were relevant in understanding the context and a particular routine, and what might not have been relevant at that particular moment, possibly was so in the near future. This aspect was especially highlighted in the discussions on whether to build the applications on smartphones only and wait for the situation to change (e.g. Interview 11) or alternatively provide a version of the application that would work on basic phones as well (e.g. Interview 20).

6.1.1.4 Alternative Explanations of Mechanisms Affecting Application Ideation

In addition to capturing local routines and using them as a basis for the applications, alternative mechanisms could also have been used. One of the clearest one would have been to simply copy the ideas for the applications from elsewhere, and to some extent, this also was claimed to be happening by some interviews (e.g. Interviews 9 and 22). This was also seen as a reason for failure, so therefore it is unlikely that copying as mechanism would have led to success, especially as many of the start-ups highlighted the need to understand the problem first and the importance of interacting with the users (e.g. Interview 31).

MECHANISM	DESCRIPTION
Local Routines as Ideas for Applications	The start-ups/developers have a perception of a particular routine and its performance. This routine is dis-assembled into different stages, and the stages are given particular attributes, which highlight how they could be improved. One or more of these stages are then turned into an application with the aim that the application on its part is able to improve the routine performance. In the process, other stages of the routine may also undergo changes, although those changes are not necessarily technological.
Copying Existing Applications	This often seemed to result in a failure of the start-up. However, external, already existing applications gave developers examples how particular problems had been solved elsewhere (similar to technological narratives).

Table 6.1. Mechanisms Affecting Idea Generation for Applications

What is problematic with this notion is that it seems to imply that an application developed outside Uganda could not be successful in the Ugandan market, and clearly this was not the

case. People with smartphones were familiar and used applications such as Facebook or WhatsApp, as was pointed out in many of the interviews (e.g. interviews 20, 55 and 56). Although it is difficult to say for sure, this difference between adopting external applications and local applications seemed to be linked to the perceptions that applications created elsewhere were automatically better that the local ones (e.g. interviews 24 and 54). As a result, this also meant that it was more difficult for a Ugandan start-up to convince their targeted users to use the app, and if the application was simply a copy of something that was already out there, it was very unlikely that it would gain large amount of users.

Table 6.1 presents the possible mechanisms behind the ideas for the applications. Although the mechanism on copying is largely rejected, it has some merit especially in relation to the role of technology in the development of these applications.

Overall, the identified mechanism for getting ideas for applications explains how a certain routine is reconstructed so that it works as a basis for the application and also where technology is situated in the process of reconstructing the routine. However, it says little of why certain routines were picked over others for the basis of application development, and although, as noted above, this can be quite difficult to pinpoint for certain, there were particular additional mechanisms that affected the process of choosing a routine suitable for an idea to build an application on. These mechanisms had different foundations in the sense that some of them were more linked to the surrounding contextual factors whereas others had a more technological foundation. In the following section these other mechanisms that affected the choosing of the routine as well as developing the application will be analysed in more detail.

6.1.2 Technological and Societal Factors Impacting Application Development

Not every routine was suitable to function as a basis for application building. Different technological and societal mechanisms played a role in influencing the start-ups to choose particular routines that were more appropriate for application development than others. Although most of these mechanisms had their biggest impact in choosing appropriate routines, they also affected the actual application building stage and even the start-ups business models and application updates after the actual launch.

These mechanisms are discussed here, and they are divided into three separate categories on the basis of the area they were seen mainly stemming from, technology, social factors or institutions. Although here discussed separately, it is noteworthy to keep in mind that in many ways these areas are overlapping, and in reality it would be difficult, if not impossible, to keep these entirely distinct from one another.

6.1.2.1 Technological Factors

Although copying and imitating could not really explain how the start-ups got the ideas for their applications, they did play a part, yet in a manner that surfaced only later on in the idea development phase and was linked to a technology's ability to solve perceived problems in the chosen routines. When a particular routine was chosen as a basis for an application, other, usually well-known applications developed elsewhere provided models to the local developers on how applications can solve particular problems. As noted above, once the local developers had chosen a local routine, deconstructed it into stages and given attributes to the stages, the question that remained was how technology could be used to solve the identified problems, and the answer to that was influenced by other applications. This was reflected in the interviews when comparisons were made by the start-ups between themselves and foreign applications like Uber or Facebook (e.g. Interviews 10 and 20).

As noted, this kind of effect became visible only later on in the idea refinement process, but it seems safe to assume that it also affected what kinds of routines where chosen in the first place. If a routine and its problem area had no room for a technological, application-mediated solution, then it was likely not chosen as the basis for an application. These models for using applications to solve particular types of problems came largely from the already existing applications, and as a result, routines that contained a problem that could possibly be solved by using technology similar to Uber for example were put forward (e.g. Interview 4). This did not mean that the entire technological base of an application like Uber was imitated (and this would have hardly been even possible for the local start-ups), but some functional aspect of the application provided ideas on how to go about solving the problem by using an application such as mapping.

In a similar manner, in addition to providing models on how to solve problems, technology's ability to provide solutions to the identified problems was also linked to the affordances it provided to the users. As noted in chapter 3, the technological resources used by developers, such as SDKs and APIs, not only enabled certain things to be built but also dictated what could not be done. Even if there was no pre-existing model in the form of an external application to solve particular type of problems, it was still possible for the developers to come up with a new model to solve a problem as long as the overall technological resources afforded that to the developers (Interview 18). However, not every routine captured a problem or a need or want that could be addressed by using an application. This can be demonstrated by the example of building an application to detect cervical cancer, and how it turned out impossible to build a mobile application capable of testing cervical cancer in an instant (Interview 23). However, it was possible to build an application that was able to give results based on ph-value instantly on the probability of an infection, which was one likely cause of cervical cancer.

As a result, technology guided the choosing of routines suitable for application development by providing certain models in the form of external applications on how to solve particular type of problems, and also by putting limitations on what could be built overall. Other areas where technology had a role in deciding whether a certain routine could be turned into an application were linked to the prevalence of a technology among the targeted users (e.g. Interviews 3 and 10) and to the issue on whether a technological resource was available for the developers to use (e.g. Interview 21). What set these two mechanisms, prevalence of technology and resource availability, apart from the technological narratives on problem solving and technological affordances was that they usually manifested themselves at a slightly later stage than the narratives and general technological affordances. However, all of them affected the suitability of a routine to function as a basis of an application.

If a particular technology, such as smartphones, was not prevalent among the targeted users, it made less sense to build applications that were dependent on the existence of that technology. Therefore, building a smartphone application that was supposed to be used by the people who did not own smartphones led companies to establish either alternative strategies or operating models (e.g. interview 10), or to discard the application idea all together (e.g. interview 7). Similarly, if the application was dependent on using for example a payment method that was not available[13], the application was unlikely to be successful. These two mechanisms were also closely related to the targeted users or customers. Some references were made that it made more sense to build applications for corporations than to individual users, as corporations already had technological structures in place that could be used for the applications (e.g. interview 22). An example of this was data: in most cases data owned by companies was often already in digital form, which the start-ups could then use as a building material for their applications. However, with individual users this was less likely and overall there were less these types of technological resources available. As a result, the technological basis on top of which the applications had to be built was lot thinner, and in most cases there were less if any already existing and locally produced technologies in place that the developers could tap on.

Overall, Corral et al.'s (2014) notions of evolving and inherent constraints (section 3.2) can be applied to describe the role of technological affordances in the studied applications. The technological constraints in mechanisms like availability and prevalence of technology were seen as largely evolving constraints, which would disappear over time. The problem was however nobody was entirely sure when this would happen or even if it would happen at all,

[13] As an example, mobile money integration to the applications was quite difficult and not available for all the developers at the beginning of the data collection phase yet became later on available for most of the developers (discussed in section 5.2.4.3).

which meant that the start-ups made more or less educated guesses of the situation, but no absolute certainty existed. Inherent constraints were more present in the mechanism regarding technological enabling and controlling, which was more taken as was without major expectations for those affordances to rapidly change. The technological narratives mechanism was different in that sense from the others, as it merely guided affordance perception by making certain solutions more visible over others.

In sum, all four of these mechanisms guided the routine selection process from a technological perspective, although from slightly different angles. Table 6.2 lists the four mechanisms, the type of filtering that occurred due to the mechanism in relation to the existing societal routines, and an example of each mechanism from the findings. The different mechanisms are to some extent overlapping, and for example for an individual developer, there was not much difference from a practical point of view if a technology did not afford a particular functionality in an application (technological resources) or if it was simply that the technology was not available for the developer in that particular location (availability of technology).

TECHNOLOGICAL MECHANISMS	DESCRIPTION	EXAMPLE
Guiding Technological Narratives	Narratives that came with the technology and the ideas on being a tech start-up that model how to solve problems by using applications	Uber-, Twitter-type of applications
Technological Enabling and Controlling	The technological resources enable but also constrain application development and place the ultimate limits on how an application is built and how it can function	Inability to create an application that would test certain diseases instantly
Availability of Technology	Technology for certain functionalities was theoretically available and possible but not present at the particular location (at the time).	Unavailability of mobile money integration (up till recently)
Prevalence of Technology	Similar to availability of technology but in this case an existing technology is not prevalent among the users (e.g. smartphones)	Smartphone penetration (no smartphones among targeted users)

Table 6.2. Technology-Based Mechanisms Affecting Routine Selection

Overall, whereas the general affordances provided by technology and availability of technological affordances were more linked to the possibility of building an application in the first place, the other two, technological narratives and prevalence of technology in the user space did not inhibit the building of an application but posed certain conditions on how the applications would be received by the users. Those also set the basis for what kinds of

technologies the start-ups chose to adopt, and how complex the applications could be. All of these four mechanisms could be seen as technological filters that a perceived routine had to pass in order to be considered suitable for an idea of a particular application. However, also other type of mechanisms existed that affected the choosing of a routine for application development.

6.1.2.2 Societal Factors

In addition to mechanisms that had a technological foundation, certain societal factors also functioned as mechanisms that affected what types of routines were chosen for the basis of application development. Like the above-discussed technological mechanisms, these societal mechanisms created filters for the existing societal routines that they needed to pass to be seen suitable for providing a basis for application development. Although it seems safe to assume that there existed a variety of these types of mechanisms, most of them were quite subjective by their nature and as a result, were highly dependent on the start-ups themselves whether they used those as filters for possible ideas for applications. However, some of these mechanisms that had a more societal basis seemed to be widely recognised by most of the start-ups, and therefore could be seen as rather common in this particular context.

The first one of these mechanisms was linked to funding and monetization. As the funding was scarce, and there were hardly any investors (e.g. interview 16 and 38), this also meant that the in order to develop the application some sort of financial resources had to be obtained. To start with, this often meant that the developers and start-up owners themselves possessed some assets or savings (e.g. interview 19) and had been able to obtain a degree (e.g. interview 18). Therefore it was unlikely that a person who was poor could become a tech entrepreneur. Also for the developers themselves it meant that in most cases they had to have additional income sources, especially in the beginning when the applications made practically no money. Many of the developers were working simultaneously for other companies (e.g. interview 8), which also meant that there was less time available for the development of their own application.

In addition, as there was little funding available, it highlighted the monetization aspect of the applications. In general there was a view that especially regarding mobile applications, nobody would pay for the applications themselves, and as a result, the monetization had to happen in other ways (e.g. interview 20). Although some start-ups opted for trying to get advertisements, in most cases this need for instant monetization meant that the routines, which were chosen to form the basis for the applications, had to include some sort of exchange of money at some part of the routines' performance. In practice this meant that applications were more likely to be

created around routines that had a payment aspect in them (e.g. interview 3), such as paying for the ride for the motorcycle taxi or for the washing of laundry.

In other words, societal routines that did not meet these conditions were often deemed unsuitable for application development. This need for monetization without the possibility to build user base first without generating income meant that also routines that might have passed the above discussed technological filters were unlikely to make it into applications. This did not lead to automatically discarding the applications development that had a more social objective. As noted by some interviewees (e.g. interviews 24 and 41), there was more money available for these type of approaches in the form of hackathons and some NGOs giving out funding to these companies. However, this type of funding was often limited, and did not necessarily result in sufficient funding for companies to be able to also choose routines where the monetization aspect was less clear. Furthermore, funding given by entities also came with conditions, such as that the application had to be developed to particular platforms or tied to the services of certain operator (e.g. interviews 4 and 23) or by dictating what kind of application had to be built (interview 17).

Related to the monetization, it also followed that a routine had to be shared by a large enough pool of people. If a routine was only performed by a relatively small group of people, it was seen as unprofitable to be addressed by the start-ups, even in the case that it had a part where monetary transaction occurred (interview 3). In a similar manner, if a user group was too poor, it was difficult to see them as a targeted user group, because only very little money, if any, changed hands in the routine performances of the poor (interview 24). However, platforms provided a bit of an exception to this, as in those the poor could be seen as a source of offering a particular service such as washing laundry.

Another mechanism that functioned as a filter for routine selection was the choice of language for the application. In a country like Uganda that has many regional languages, English that was one of the official languages of the country was often the choice for the applications' language, and only one or two start-ups had even thought about incorporating local languages into their applications. The general view was that it made little sense to build application using local languages, as the start-ups saw that their targeted users spoke or commanded at least a certain level of English (e.g. interview 6 and 18). It was unclear whether this was actually the case, but similarly it meant that routines that existed among groups that did not speak English would not have been seen suitable for the basis of applications. As a result, the lack of English skills among a particular group also meant that their routines were unlikely to be chosen for application development.

In addition to the technological mechanisms, these two mechanisms functioned as filters when selecting routines on top of which the applications would be built upon. Table 6.3 lists these two mechanisms. As noted, several others also likely exist, yet it was difficult to have certainty on how common those were across start-ups and also the subjectivity of the start-ups regarding those other mechanisms tended to be quite high, where only few interviewees mentioned them while others either did not mention them or saw them as being non-issues.

SOCIETAL MECHANISMS	DESCRIPTION
Issues Related to Funding	Chosen routines had to have a part which included monetary transaction that enabled the applications to instantly monetize.
The Position of English Language in the Society	Develop applications whose targeted users understand English

Table 6.3. Societal Mechanisms Affecting Routine Selection

6.1.2.3 Institutional Factors

In addition to the technological and societal factors, there were also certain institutional factors that affected the type of routines that were chosen for the basis of application development. Although these were heavily linked to the general societal issues which Uganda faced, they are here dealt separately from the societal factors as their foundation was closely linked to certain institutions that were in place in Uganda.

As discussed, in many respects Uganda is considered a developing country, which then provides a particular type of context not only for application development but also more generally. One way that this had an impact on application development was through the institutions that had been set up in the location. One notable example of this was the large number of NGOs and international development agencies that had a presence in Kampala and beyond (Fieldnotes 3). As noted above regarding the need for instant monetization, some of these institutions displayed themselves as possible sources of funding for certain type of applications (e.g. Interviews 16 and 23). In other words, there was an incentive to create applications that had a developmental objective built into them, as this enabled the start-ups to possibly tap into financial resources that otherwise would have been unavailable for them.

Like the developmental organizations, the tech hubs had an impact on what type of applications were created by the companies that resided in them. As noted, these tech hubs were often branded in a particular way, and whereas one hub was concentrated more on start-ups that were purely for profit, another had an objective to incubate companies that in addition to making profit would try to accomplish certain social goals as well (Fieldnotes 4, Interview 41). To some extent, the whole narrative of application development appeared to be linked to the notion of how applications could have a positive impact on societies and solve societal

challenges. Similarly, the narratives that came with the technologies and stemmed from the successful applications created elsewhere had an impact in developing the applications locally, as these provided examples on how the applications should work and therefore also what kinds of applications could be created. In addition, the organizations that provided developers the technological resources, such as Google with Android, had an impact on how the developers saw themselves, i.e. more as tech start-ups than entities trying to solve local challenges, which also possibly impacted the problems in routine perceptions regarding the application development (Interviews 28 and 44).

All of these, the NGOs, tech hubs and well-known external application companies, had an institutional basis and overall they impacted what kinds of routines were chosen for applications and how the applications were built. Table 6.4 summarises the mechanisms identified here that stem from the institutional factors such as international technology corporations and other local organizations that are present in the location where the applications are developed and launched.

INSTITUTIONAL MECHANISMS	DESCRIPTION
Preference Setting by Local Institutions	The surrounding institutional settings set preferences on what type of applications are preferred over others.
Image Setting by Technology Corporations	Technological resources and the entities behind them provide start-ups models on how to solve particular challenges as well as what a tech start-up should look like.

Table 6.4. Institutional Mechanisms Affecting Application Development

Figure 6.4 displays where in the application development process the identified filters are placed. Although the figure places them in the perception stage, it did not automatically follow that it was at the routine perception where unsuitable routines were filtered out. In some cases the filters or mechanisms that affected routine selection for application development became visible for the developers and start-ups only at a later stage such as after launching the applications. It was in those stages when the start-ups became aware of their impact, which also led to altering the existing business models or functioning of the applications. However, for most routines their suitability for application development became apparent already at the perception stage, although as noted above, the way routines were perceived by start-ups did not always match with the ones of their targeted users.

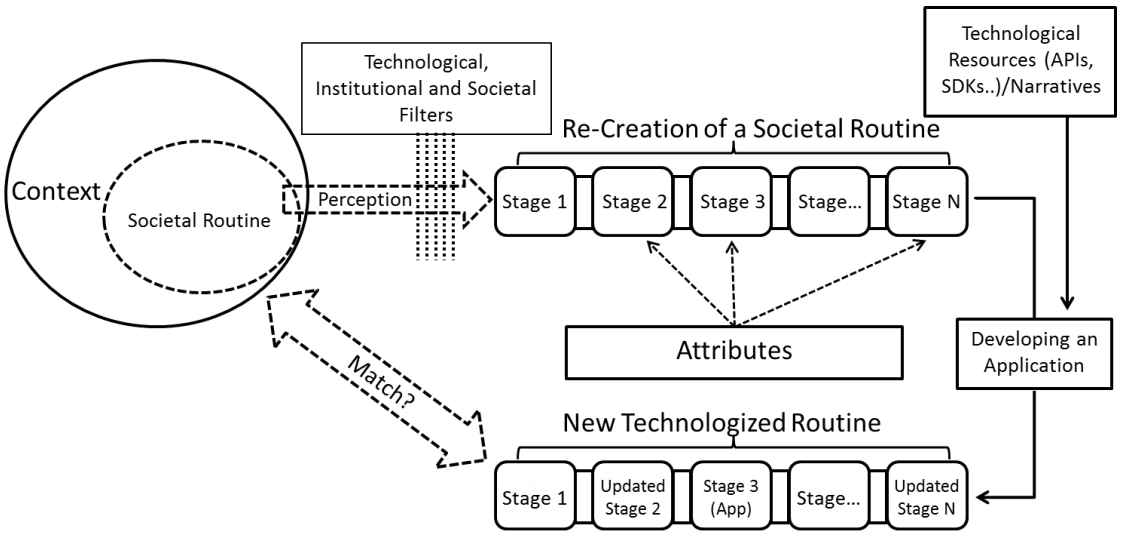

Figure 6.4. Technological and Societal Mechanisms as Filters for Routine Selection

6.1.2.4 Alternative Mechanisms for Explaining Routine Selection and Application Development

As noted in the beginning of this section, there were several mechanisms that appeared to have a role in deciding which routines were chosen over others. However, the ones presented here have in common that they seemed to be shared and experienced by relatively many of the interviewed start-ups and therefore presented here. If the analysis would concentrate on one start-up alone, it might very well be that other mechanisms would have surfaced and played even a more important role than the ones discussed above. However, on a larger scale those mechanisms would not have been shared by many other start-ups.

When the applications were finally launched or interaction with the targeted users had begun, it was then that the accuracy of the perceived routines and the identified problem (as well as the solution for the problem) became crucial for the success of the applications. Similarly, if the routine had been inaccurately picked overall in a manner that it did not pass the filters but was turned into an application all the same, it was likely to get a lukewarm reception at best from the targeted users.

6.1.3 Outcomes of Application Development

Summing up, the applications were based on particular perceptions of routines that existed in the local society. Those routines had to pass certain technological, societal and institutional filters in order for them to be considered appropriate for application development. Once a routine was chosen, the routines performance was dismantled into stages, and within those stages, unmet needs and wants or problems that needed solving were identified by giving these stages particular attributes. To address a need or want or to solve a particular problem, an application was built around it, which was then further launched to the targeted users. The

question was however whether the perceived routine matched the perceptions of the targeted users, and furthermore, were the views of the identified problem area shared by the targeted users and did they agree with the proposed, technologically mediated solution. This was crucial for the applications to succeed, and usually the answer to these questions came at the time of launching the application which we shall analyse next.

6.1.3.1 Interaction Space between Existing Routines and Applications

Following the model described in Figure 6.4, it was at the time of launching the applications that the applications and the proposed new routine performances the applications carried with them were fed back to the targeted users. If the targeted users accepted the new technologically-mediated form of performing the routine, i.e. the new routine was able to replace the existing routine, the new routine formed a part of and impacted the existing context and contextual factors. An example of this could be for example using an application to order a motorcycle taxi or use a helmet when taking motorcycle taxis, which would have meant changes to certain parts of or even replacement of the existing routine and its performance.

It was not given that existing routines were to be easily replaced or even slightly modified by the applications. To begin with, the general perception of locally produced applications was not very high. Many of the targeted users were not convinced that the local applications can provide them significant improvements regarding their routine performances (e.g. Interviews 47 and 55). Another issue was that often the perception held by the start-ups of an existing routine did not match the views held by the targeted users. This was even further enhanced by turning at least part of that routine performance into the form of an application, which deviated the performance of the routine even more from the original one.

In addition, it did not automatically follow that the existing routines were somehow inferior to the proposed new ones or in general not functioning well. These societal routines had evolved over time and were also dynamic, which meant that they had adapted to changing contextual factors. This does not mean that there was nothing to improve, with or without the use of technology, but that in many aspects the existing routines fitted the local context well, especially regarding certain parts of the routines. As an example, although the motorcycle taxis were known to be unsafe, they were also very convenient and quick, and one could usually get one almost instantly. An application that offered a way to have them come to you did not necessarily improve convenience, and possibly made the waiting time of a motorcycle taxi longer (e.g. interview 24).

Overall, it was not surprising that the targeted users were hesitant towards the applications and the routine changes that came with them. Rogers (2003) for example discusses the five factors

that affect how likely it is for a user to adopt an innovation. Among those are items such as how much advantage the innovation is seen as bringing to the user, how complex it is to use, or how consistent the innovation is with the existing values and experiences of the users. To some extent, already the use of technology made the adoption of the application less likely in some cases. Furthermore, the perceptions that the start-ups themselves had of a particular routine often differed from the ones held by the targeted users, who were also the actual performers of those routines. As a result, even though the start-ups and the users shared the same location, the perceptions of the same routines remained different, which also meant that the start-ups that paid more attention to studying the routines before building the application in general tended to fare better.

As mentioned, when targeted users tried the applications, that also functioned as a verification for the application developers on how well the perceived routines and the proposed solutions to the problems that existed within those routines matched with the views held by the targeted users. If there were major discrepancies between the two, it was unlikely that the application would gain success among the users, at least not without changes either to the application or to some other aspect of the new, proposed routine. Furthermore, in many cases the parts of the new routine that seemed to cause most problems for the start-ups were the ones that were supposed to be dealt with by using technology, i.e. the application (e.g. Interview 3).

6.1.3.2 Divide in Existing Routines and Technologically Mediated Routines

Overall, due to inaccuracies in routine perception by the start-ups, the new, technologically mediated routine either did not make sense to the users or then the identified problems or needs and wants were not shared by the targeted users (e.g. Interview 18). However, in many instances the part of the technologically-mediated routine that appeared to inhibit success among the targeted users was the one linked to technology (e.g. Interview 16). Even if the routine was perceived correctly and the users agreed on the identified problem, the applications themselves appeared to be the reason that impeded adoption.

This could partly be traced back to the lack of certain technologies among the targeted users. As many start-ups noted, it was difficult to reach certain target users because they did not have smartphones, which also meant that in some cases the applications had to be designed on basic phones. This on the other hand had the effect that many of the functionalities had to be left out from the application, such as using maps to pinpoint locations (Interview 3). The other option was to settle for smaller user groups, or to simply hope that with time the ownership of smartphones would increase and the target user groups would grow automatically (e.g. interview 12).

Lack of technology led to increase in manual work for the start-ups, since as certain functionalities like payments or support functions (Interviews 3 and 10) could not be automated by the start-ups, they had to remove these functionalities or do them manually for example over the phone by talking with the users. However, this also indicated that the problem that the start-up had identified was most likely shared by the users, and as such, the recreated routine managed to meet the expectations from the users. The side effect was though that the increasing amounts of manual work also made it more difficult to scale up, because that would have demanded additional resources in terms of new employees that the start-ups could not afford.

However, lack of technology among targeted users only explained part of the problem. In addition to this, the solution models offered by the technological resources did not necessarily fit the existing routines overall. Examples of these were for example the websites that used online stores only as catalogues without permitting ordering or even reserving the items over the web (Interview 33), or the possibility to order a motorcycle taxi by using an app that had difficulties in gaining attraction among users (Interview 10). In these cases it was not so much about lack of technology, but more that the technology and the proposed technological solutions did not fit with the existing routines. In other words, there was a mismatch with the problem and the solution put forward by the technological resources. This aspect got further highlighted as many of the members of the start-ups were programmers and therefore predetermined to provide a technological solution to the identified problems. To some extent it appeared that no matter the identified problem or the need or want that was to be addressed, the answer for it had to take the form of an application or some other technological tool.

Díaz Andrade and Urquhart (2010) found in their study on ICT implementation in rural Peru that *"existing social networks and the superimposed technological network were incompatible"* (p. 370). To some extent the same occurred between the existing routines and the new, technology-mediated routines proposed by the start-ups, especially in relation to the parts of the new routines that included the usage of application. This was partly due to the narratives that came with the technological resources on how certain problems should be solved, but also in the form of the affordances of the technological resources that dictated what should be done and how. These technological solution models did not necessarily fit with the existing local routines, which led to a mismatch between the technology and the routines that were chosen to form the basis of the applications.

Another way of explaining the differences between the applications and the existing routines can be done by using the notion of ostensive and performative aspects of routines (Feldman &

Pentland, 2003). The applications became the ostensive aspect of the routine that described how a particular routine should be performed according to the developers and the start-ups. What is noteworthy is that this ostensive aspect was describing the re-created routine, and not the existing way of performing the routine. The performative aspect of routine, i.e. carrying out the routine according to people's understanding of the routine and its purpose, was more linked to how the routine was at the time performed by the targeted users. In this sense, the notion on the ostensive and performative aspects of the routines was in a certain manner reversed, where the ostensive did not actually describe the existing routine but what the routine was hoped to become by the developers. If the application, or the ostensive aspect of the new routine, was accepted by the targeted users, then this ostensive aspect of the routine took its place as actually describing the routine as it was performed. However, what often happened was that the ostensive aspect was not accepted by the targeted users, who kept on performing the routine as they had done so far without incorporating the application into it (e.g. Interview 8), and thus in a certain manner rejecting the ostensive aspect of a routine suggested by the developers and with them, the start-ups.

As discussed in chapter 3, the ostensive and performative aspects of routines are in relation to and shape each other. In a similar manner, the user feedback formed an integral part of shaping the ostensive aspects of the proposed routines by making the start-ups more aware on the performative aspects of the existing routines. As a result, the ostensive aspect, i.e. the application, and the performative aspect, as in the existing routine, resembled each other more as the user feedback was incorporated into the application. However, the ostensive aspect could not become entirely similar to the performative aspect of the existing routine, because that would have meant that the routine had stayed the same and no application would have been incorporated into the performance of the routine, and no betterment for the existing routine would have taken place. The same was not true though vice versa, since if the performative aspect became the same as the proposed ostensive aspect, it meant that the application was used by the targeted users and they had accepted the re-created routine proposed by the start-ups, which was also the main objective for the start-ups themselves (e.g. Interview 18).

In sum, the applications could be seen as ostensive or descriptions of how a particular routine should be performed according to the start-ups. This ostensive aspect of the routine is conditioned by the start-ups' perceptions of the routines that their applications are based on, the institutional, societal and technological filters affecting the choosing of the routines, and by the technological affordances and technological narratives that come with the technological resources. When this ostensive aspect is then offered to the targeted users, they get to decide whether to accept it and if so, the application becomes a description of how a routine is

performed. As will be discussed next, if the targeted users use the application, it becomes a source of technicity in the daily transductions of the targeted users.

6.1.3.3 Space and Code

As discussed in Chapter 3, Dodge and Kitchin (2005) discuss the different ways software or code affects the everyday lives of people either in the form of coded objects, coded infrastructures, coded processes or coded assemblages. These four categories can be seen as incremental as they presuppose the existence of the previous category before making possible the following one. In other words, it is difficult to have coded infrastructures, at least functioning ones, if there are no coded objects. Every category therefore relies on the existence of the category below it, and is built partly on the affordances that the underlying category provides.

In relation to this categorization, many of the new, technology-mediated routines could be seen as intentions to create coded processes. Dodge and Kitchin (2005) define coded processes as the transactions and flows of digital data that travel across coded infrastructures, which on the other hand are the networks that link coded objects. As discussed above, in the application development process routines, or parts of routines, were given a technological form by building applications around them. These applications based their technological functioning on data that was collected, processed, transformed and delivered to another location. However, due to the lack of coded objects and infrastructures, this type of data was not always available (e.g. interview 46). It became increasingly difficult to meet the overall objective set for the applications to automate certain parts of the selected routines, as there was little basis to do these types of automations since the existing routines did not contain technologies that would have enabled this (e.g. Interview 10).

As a result, there was very little technology on top of which the functioning of the applications could be built on. Another way to describe this is to say that the level of technicity in many of the local transductions was not very high to begin with. Using these notions of technicity and transduction deployed by Dodge and Kitchin (2005), the existing routines, which can be seen as transductions, were not usually dependent on digital technologies in their performance. Therefore, the applications were often the first layer of technicity that was being tried to incorporate into the local routines. However, these applications could not rely on any other type of code already present in the routines, be that in the form of objects or readily available data. In other words, although the applications themselves could be assembled by using external technological resources, the local technological resources that would have enabled users to gain more benefits from the applications were scarce (e.g. Interview 22 and 30). As a result, the applications themselves had to provide much of the needed local technological resources if

those were essential for the functioning of the applications. Certain local resources were available, such as access to network and some computational capabilities provided by the devices used to access the applications, but there were not too many additional, local resources to plug into. This also had an impact for the solutions provided by the applications, because some of the technologically possible solutions were not available in the transductions that occurred in this particular location, as the technicity of those transductions was at a level that it was unable to provide the developers these resources (e.g. Interview 3).

As mentioned, if those applications were successful and became part of the transductions of their targeted users, in many ways these applications would have provided the first levels of technicity into performing those transductions. Since these applications could open APIs for other companies to use, it might make it easier for the next wave of applications to tap into this technicity that would already exist in the transductions (e.g. Interview 22). As a result, this would have helped the developers to create applications that would have formed a part of what Kitchin and Dodge (2011) refer to as code/space or coded spaces. However, it can be difficult to achieve this with transductions that have little if any technicity built into them. This also means that in relation to many transductions, the applications were not essential for carrying them through, which also had implications for the applications' ability to solve identified problems or address existing needs and wants.

As a practical consequence of this for the start-ups, the lack of local technological resources led to the simplification of the applications or even fading them more into the background in the business models of the start-ups (e.g. Interview 4, Fieldnotes 8). However, it is important to note that this was not only about access to technology by the targeted users, or about their ability and skills to use certain technologies. Those factors naturally had an impact as well, but the issue was also related to a large extent on the existing societal routines within this particular location. As the performance of those routines relied very little on any kind of information or communication technology, it also meant that the new, technology-mediated routines had very little to base their functioning on, and as a consequence the applications were unable to gain traction among the targeted users. In addition, the existing routines served mostly their purpose, and from the perspective of the users, there was very little to be gained from the solution models proposed by the technologies used for application development.

Naturally, exceptions also existed, but the applications that had had more success relied on routines where technology had already a role (interview 18), and in that sense were more appropriate for technology-mediated problem solving, or then those were applications that functioned more as standalone items such as games, with less need for categories like coded

infrastructures. The former type of applications also shows why it made more sense according to some start-ups to target entities like corporations, as the requirement for coded objects and infrastructures was more likely to be met regarding the routines of those organisations (Interview 22).

In relation to the concept of code/space discussed in Chapter 3, the lack of technology in routines meant that it was also more difficult for the start-ups to create applications that would have formed part of code/space or coded spaces (Kitchin & Dodge, 2011).. As the existing routines had very little technology incorporated into them and did not usually rely on coded objects or infrastructures, also the applications were unlikely to become essentials for performing certain societal routines. This in practise meant that either the application had to be changed or dropped altogether as a way to provide a particular service.

MECHANISM	DESCRIPTION
Lack of Existing Technological Layers	Makes it more difficult for developers to develop applications on top of existing technologies, which would also make it easier to build applications forming part of code/space or coded spaces, and as a result, it is relatively simple for the users to revert to the existing routines.
Mismatch between Developer and User Perception	The new, technologically mediated routine performance deviates considerably from the perceptions of the targeted users due to inaccuracies in routine perception and technological solution models that do not fit the local context.

Table 6.5. Mechanisms Affecting Application Launch and Usage

Table 5 summarizes the mechanisms affecting the launch and usage of the applications. In sum, most of the existing social routines were not suitable in their current state to form a basis for application development. However, as noted above, this did not necessarily mean that the reconstructed routine and the suggested improvement would not have been agreed by the users, but in order for the new routine to be accepted it had to be stripped away from the technological parts. In other words, it appeared that many of the technologies that were used were not suitable for the tasks they were given in the reconstructed routines.

This is naturally something that is subject to change over time as technologies get more incorporated into the existing routines. As noted, context and contextual factors are dynamic by their nature, and the same applies therefore to societal routines. However, especially in relation to applications that required the usage of more advanced technologies, the leap from the existing routine to a new, technology-mediated routine was too long. The technological resources provided for applications appeared to be largely unable to get incorporated into the local routines and provide improvements within them. For the start-ups this meant the reasons they had been set up i.e. to create applications in order to reconstruct existing societal routines and solve certain problems, were unattainable as the principal tools to achieve these objectives

were not adequate for the task, at least not at the time when data for this research was collected.

Another solution for this was to reach users that were better positioned to use the applications already by having access to the more advanced technologies such as smartphones. However, this often came with the price of decreasing numbers in usage, although also had the benefit of enabling the start-ups to automate certain areas of the re-created routine and cut for example labour costs. Another option was simply to resort to less advanced technologies, such as basic phones, which were already familiar for the targeted users. However, this also meant that there was an increase in the areas that had to be done manually for example by calling. Either way, neither of these worked if the newly created routine behind the application had been erroneously perceived by the start-up. From this perspective it was also interesting how the technology hubs in the area, as the name implies, were largely technology driven and offering a lot of training regarding different technologies. What appeared to receive less attention was training linked to understanding the problem spaces, i.e. the needs and wants of the targeted users and the problems they encountered in their daily lives. As a result, it often seemed that for many of the application developers the answer for almost any problem was technology in one form or another, despite the fact that the transductions of the targeted users often provided little fertile ground for this kind of approaches.

6.1.3.4 Alternative Mechanisms for Explaining Application Usage

It is also possible that application usage was affected by other mechanisms. One of those could be related to functionality of the applications, in which case usage would be dependent on the overall quality of the applications. Another mechanism could be that the result of application adoption is largely dependent on luck and timing, and with a bit more time the applications would have been able to attract users in greater numbers.

Both of these mechanisms have some truth in them, yet they are not entirely convincing either. If the applications had completely failed to function, it is likely that the developers would have stopped building them before launching them and as a result, the applications would have never even entered the market. While true that some of the launched applications were not that complex and lacked certain functionalities, their usefulness was still evaluated from the perspective of how beneficial or entertaining they were and less how well they functioned. In other words, most judged the applications on the basis of whether they saw them as adding value when compared to the existing way of performing the routines.

The same applies to the notion of luck or the issue of timing. It did occur that applications were seen as beneficial in ways that the developers had not expected (e.g. Interview 8), but even then

the reasons for their usefulness were connected to the factors regarding what the applications allowed their targeted users to do. Timing was naturally an issue, and as discussed in section 6.1.2., certain functionalities could not be built because of the necessary resources were not available at the time. However, even though certain resources were not available, the applications were still based on currently existing routines and different timing might have meant that the routines would have been different, and therefore timing could not explain the variety of outcomes in application usage alone.

6.2 Conclusion

Table 6.6 summarizes the mechanisms found in the analysis. It also shows at what stage of application development each mechanism had most impact and what is the foundation where the mechanisms stem from. These mechanisms provide answers to the research questions that were set in chapter 3, which as a whole answer to the main research question of this research, "*how does context impact the application development process in a developing country?*" As noted, the identified mechanisms function as the key mechanisms that were common across a large number of the studied applications and start-ups. Other mechanisms also played a role, but their impact was often limited to one or two of the applications or they were only relevant to particular type of applications like games or health related applications.

Starting from the first sub-question set for this research *i.e. "how do societal routines display themselves in the application development process?",* the answer lies mainly in the role of these societal routines as sources of ideas for the applications. The applications were based on the routines that existed in the society and formed a part of recreating the routine in a manner that would somehow improve some area of the existing routine by using technology, i.e. the application. Furthermore, certain societal issues like existing flows of money and usage of language had implications for what kinds of routines were selected for the applications as well as on how the applications themselves were designed. In addition, the organisations that were present in the location and interested in particular types of applications influenced the types of applications that were developed by organizing for example hackathons with the objective of promoting application development linked to a particular area such as education and health.

In a similar manner, the technology hubs where many of the application development start-ups resided often placed emphasis on certain kinds of start-ups that were taken into the hub, for example by portraying themselves as hubs who also aimed to promote social objectives in addition to making profit.

MECHANISM	DESCRIPTION	STAGE IN APPLICATION DEVELOPMENT	FOUNDATION
Local Routines as Ideas for Applications	The start-ups/developers have a perception of a particular routine and its performance. This routine is dissembled into different stages, and the stages are given particular attributes, which highlight how they could be improved. One or more of these stages are then turned into an application with the aim of improving the routine performance. In the process, other stages of the routine may also undergo changes, though those are not necessarily technological.	Source of ideas for applications	Context (local routines)
Copying Existing Applications	This often seemed to result in a failure of the start-up. However, external, already existing applications gave developers examples how particular problems had been solved elsewhere (similar to technological narratives).	Source of ideas for applications	Technology
Issues Related to Funding	Chosen routines had to have a part which included monetary transaction that enabled the applications to instantly monetize.	Filtering ideas	Context (Societal Factors)
The Position of English Language in the Society	Develop applications whose targeted users understand English.	Filtering ideas	Context (Societal Factors)
Preference Setting by Local Institutions	The surrounding institutional settings set preferences on what type of applications are preferred over others.	Filtering ideas	Context (Institutions)
Image Setting by Tech Corporations	Technological resources and the entities behind them provide start-ups models on how to solve particular challenges and what a tech start-up should look like.	Filtering ideas and development of applications	Technology/ Context (Institutions)
Guiding Technological Narratives	Narratives that came with the technology and the ideas on being a tech start-up that model how to solve problems by using applications	Filtering ideas and development of applications	Technology
Technological Enabling and Controlling	The technological resources enable but also constrain application development and place the ultimate limits on how an application is built and how it can function	Filtering ideas and development of applications	Technology
Availability of Technology	Technology for certain functionalities was theoretically available and possible but not present at the particular location (at the time).	Filtering ideas and development of applications	Technology
Prevalence of Technology	Similar to availability of technology but in this case an existing technology is not prevalent among the users (e.g. smartphones)	Filtering ideas and development of applications	Technology
Lack of Existing Technological Layers	Makes it more difficult for developers to develop applications on top of existing technologies, which would also make it easier to build applications forming part of code/space or coded spaces, and as a result, it is relatively simple for the users to revert to the existing routines.	Development of applications/ Application Usage	Context (Technology)
Mismatch between Developer and User Perception	The new, technologically mediated routine performance deviates considerably from the perceptions of the targeted users due to inaccuracies in routine perception and technological solution models that do not fit the local context.	Filtering possible ideas and development of applications	Technology/Context (local routines)

Table 6.6. Identified Mechanisms and Their Role in Application Development

Finally, existing societal routines and their performance also impacted how the applications were received by the targeted users in terms of how well the users perceived the application to

improve the performance of the routine into which the application was incorporated. Overall, the applications became the ostensive aspects for the new, technologically mediated routines set forward by the start-ups. The intention of the start-ups was that these new routine would replace the existing routines in the repertoire of societal routines of the targeted users.

In relation to the second sub-question, *"how are technological resources chosen to develop the applications and how these resources match with the developers' ideas for the applications?",* the technological resources used for the applications were a result of a balancing act between the developers' skills and the technologies that were prevalent among the targeted users. What is noteworthy however is that even though the targeted users might not been in possession of the technologies needed to use the applications, such as smartphones, in many cases the start-ups still went on and built the applications on platforms that required these technologies. It appeared that the thinking among these start-ups was often that over time the situation would change, i.e. the users would have the required technologies. As a temporary solution, the users were also provided for the possibility to use the service by other means, for example by calling, and without the need to use the application. The choosing of the technologies was overall often impacted also by technological narratives that came with the technological resources. An example of this was the usage of the Google's Android platform, as trainings in the hubs were provided to create Android applications although the targeted users might have lacked the mobiles to use these applications. Furthermore, the chosen technologies where often chosen with the intention to automate as many of the areas as possible of the new routine. However, this turned out to be in some cases impossible as less technological solutions such as the option for calling had to be provided on the side. Despite this, the technologies seemed to match the start-ups' ideas relatively well. This was partly because many of the start-ups' members had a background in software engineering, which also meant that when choosing routines on top of which the applications were built, the developers were looking to identify problems that could be addressed by using technologies they were familiar with. As a result, the selection of appropriate technological resources for the applications was not done solely on the basis on what best fit the targeted users, but was also influenced by the technologies themselves and the narratives that came with them. These factors contributed further that the proposed new, technologically mediated routines were often not adopted by the targeted users.

This can be seen in relation to the final sub-question of this research on "*how are the applications received by the targeted users and what kinds of implications this has for the applications?"* In most cases the applications were only able to receive modest amounts of usage. The reasons for this were mainly linked to two factors: the mismatch in routine perceptions between the targeted users and the start-ups and to the limited amount of local

technological resources that were present in the existing routines. Regarding the first, what was viewed as a problem or an unaddressed need or want by the start-ups was not perceived as one by the targeted users. Thus, the proposed new routine failed to become incorporated into the routine repertoires of the users as they saw less benefit in adopting the new routine. Furthermore, due to the low levels of technicity in the existing routines' performance it was difficult for the start-ups to create applications that would have enabled the users to do something they could not otherwise do. In other words, the applications mostly failed in forming routine performances that could be seen either as code/space or coded spaces. What also occurred in some cases was that the certain stages of the new routine were accepted by the users, but the stage that was supposed to be taken care by the application was not. Overall, the applications that seemed to have more success were the ones where substantial work had gone into understanding the problem first and building the application only after this, instead of building the application first and launching it to see if it gathered any traction among the users.

The analysis done here hopes to provide a more detailed understanding of application development as a process where also the location and context impacts the outcomes of those applications. Furthermore, technology overall conditions the application development process through different narratives that come with it. This has also implications for the areas discussed in the literature review in chapter 2, especially in relation to our understanding of digital divide and technology implementation in developing countries.

7 Discussion and Conclusion

"Technology is a useful servant but a dangerous master." (Lange, 1921)

This chapter connects the results obtained from the data analysis to the literature presented in chapter 2. It starts off by discussing the notion of locality in application development occurring in developing countries and compares the research's results to the arguments made elsewhere in the literature. After this, the chapter will look at each of the stages of application development (i.e. ideation and design, application building and usage) to show how this research may shed further light to research areas linked to these stages. From there the chapter moves on to discuss how application development process can be understood from a developmental perspective. Finally, the key methodological, theoretical and practical implications of the research are presented, followed by sections regarding areas for further research and limitations of this research.

7.1 Discussion on the Key Results of This Research

7.1.1 Application Development in Developing Countries

The research findings and the analysis of those findings presented a view of local application development as a process that is ultimately linked to the existing societal routines and mechanisms that are present in the location where application development takes place. In comparison to applications created elsewhere, local development of applications has been seen as well positioned to take into account relevant contextual factors and as a result, to create applications that better match local needs and wants. As noted in the literature, applications built *in situ* are likely to be of more interest to the local users (Powell, 2001), be more aligned with local realities (Heeks and Molla 2009) and be consumed by the targeted users (D'Costa, 2006). Equally important, it is argued that problems such as design-reality gaps (Heeks, 2002b; Masiero, 2016) can be more easily avoided when the technology producers are located in the same place as the targeted users and being aware of the local context, which leads to better possibilities in terms usage and sustainability[14].

However, as shown in the analysis chapter, locality in itself did not guarantee that the routines were automatically perceived accurately by the start-ups. The original perceptions of the routines in many cases failed to match the perceptions held by the targeted users of the users,

[14] As noted in other research (Wagner & Fernández-Ardèvol, 2016), in a developing country context there is sometimes the tendency to try to create applications for other markets that are seen as more lucrative. What is noteworthy is that viewing application development as a process based on existing societal routines does not automatically mean that the key actors need to be present in the locations. It is more that the likelihood of knowing the chosen routines well is higher if the start-ups are local, since it can be difficult to observe routines accurately from a distance.

which also caused difficulties in application adoption. In many cases considerable work had to be done in order to achieve an accurate understanding of the chosen routines, and in general applications that started by studying the routine first and only then building the application tended to be more successful. This also helped the start-ups to choose the technological resources more wisely, as it was the identified challenges that dictated the building of the application instead of the technological resources, which happened in cases where applications were built first using technological resources that did not fit the purpose particularly well.

One common reason for the differences in routine perception between the start-ups and the targeted users appeared to be the different socio-economic positions of the two. As noted, the developers and other start-up personnel came from relatively privileged background as they had had the resources to educate themselves and also were able to dedicate time to developing applications of their own without guarantees of future rewards. This was not necessarily the case with the targeted users, especially if those were from more marginalised segments of the society. In a society with significant income inequalities and large part of the population living in relative poverty (UNDP, 2013), it was not granted that the start-ups were aware for example of the lives of the poor, which again would have been necessary in order to understand the routines performed by these targeted groups of users.

The results of the analysis therefore indicate that locality alone is not sufficient to meet the requirement of having good awareness of all the necessary contextual factors. Even if the start-ups resided in the same location than their targeted users, it did not guarantee that the perception of the existing societal routines was correct and matched those of the targeted users. However, this does not mean that there was no benefit by being present in the location, but more that people living in the same place but coming from different backgrounds can be quite unaware of the realities of the others. To understand completely the challenges faced by the targeted users, it was still necessary for the start-ups to interact closely with the users in order to get a better idea of the problems and also of the ways to solve these problems by using applications. However, the interaction was naturally easier by being present in the same location as the users.

In the literature, a significant amount of research has been undertaken to identify the key variables that need to be met for software industries to thrive. In addition to connectivity (Ojanperä et al., 2017), developers' skills, investments, competition between companies, clustering and government policies have been mentioned as important factors in this regard (Heeks, 2006; Kambhampati, 2002; Molla, 2000). Furthermore, socioeconomic characteristics such as education, social class, race or gender, have been seen as affecting people's willingness

to take part in content creation (Schradie, 2011). In relation to these factors, the results of this research do not negate their general importance. Connectivity was an issue not only in relation to using the application but to some extent building them as well, where for example technology hubs were rated according to the quality of the internet connection they had. Developers' skills were naturally essential for building the applications, and lack of investment and supportive government policies at the time made it more difficult for the companies to become viable. Competition between different start-ups was mostly scarce, although in areas where there was more of it such as web design some of the companies did appear to do rather well. Furthermore, as noted above the start-up owners themselves appeared to come from relatively privileged backgrounds, giving support to Schradie's (2011) argument on the link between content creation and socio-economic characteristics.

However, it is noteworthy that the success of the start-ups was largely dependent on the ability to identify users' routines correctly and propose an enhancement to the routines that was agreed by the users. As an example, even if a start-up had the skills to develop the applications and had found investors funding the enterprise, usage of the application was not always guaranteed. If the intention is to create applications or software for local markets, it is equally important, if not even more, to take into account the issues related to the chosen routines and to the performance of those routines. If the user base is ill-equipped to use the applications or the chosen routines are not thoroughly understood, applications are unlikely to be the best approach to create successful businesses. Overall, too much emphasis has gone into the technological and institutional aspects of creating local application industries, especially in cases where the initial target market for the applications is local. What seem to be missing are the tools to understand the local challenges that are to be addressed with the applications. As a result, in addition to technological skills other type of skills are also needed, such as those related to problem definition and understanding.

Although true that no matter the location, poor understanding of the problem from the part of the developers is likely to lead the application to fail, there are three factors that set developing countries apart from the more developed ones and also from themselves. The first one is that the problems that the applications are likely to be somewhat different and may result more from issues such as weak institutions, which is less likely in developing countries. Second, the targeted users might differ considerably from the developers in terms of basic technological skills and access, and overall as income differences tend to be high, it can be more difficult for the developers, who tend to come from relatively privileged backgrounds, to fully grasp the problems that some of their targeted user groups face.

Finally, and related to the second one, the technological basis that the applications are built on is lot thinner than it is in more developed countries. Even though the technological resources needed for the application development such as SDKs and many APIs would be available, there are very few local resources available in terms of local digital data or users that have access to smartphones and internet connection. What this means is that less of the different stages of routine performance can be automated and have to be treated manually or left as it is. For the start-ups, this means more manual work for example in providing alternative contact methods such as support by calling, and also that they are less able to provide fully fledged applications because their targeted users simply do not have the tools to use the services or the data that is needed to provide those services is not being provided by anyone. This further means that becomes more difficult to scale up, because that automatically means hiring more workforce to serve users who are not necessarily bringing in that much revenue to cover the costs. On a more positive note, the start-ups themselves reported that setting up the start-up and trying to turn it into a profitable business had been in itself a very useful learning process, during which they had acquired new skills and experiences that were likely to help them in the future. In addition, in some cases the start-ups had also turned out to be in some aspects successful by managing to generate revenue and also by providing tangible benefits to the targeted users for example in the form of increased incomes. However, the truth was still that there were quite a few start-ups which were unsuccessful in launching the applications or attracting users. It often appeared that even though gaps in terms of geographical location were diminished in local application development, other gaps remained and became visible during the different stages of the application development process.

7.1.2 The Contextual Basis of Application Ideation and Design

As discussed in the literature review in chapter 2, the design processes are said to be influenced by the environments where they take place, and for example general societal norms and values impact how particular products or services are designed (Bijker, 1995; Pinch & Bijker, 1984). These values and norms stem from cultural and social contexts, which tend to vary according to the location they are found in and even within the same location change over time. Ananny et al. (2007) note that in order to understand where technologies or the ideas behind them come from, it is necessary to focus on the contextual factors that impact the design decisions that have been made regarding these technologies.

Every design is also linked to other designs and design processes. In terms of design path dependencies, the literature stated that new design paths are based on either changes in technologies or on events in other areas of the design paths (Fuchs & Shapira, 2005; Schienstock, 2011). Furthermore, new design paths are not to be seen as deterministic but more as matters

of choice, as they can be based on choosing between some of the paths that exist in particular location, and in this sense follow more a logic of path generation than a dependency (Djelic & Quack, 2007). However, in a location that has little tradition in technology design, it is not entirely clear what are the relevant other pre-existing paths that impact the new design. The focus of this research, local application development, had few if any precedents in the location of Kampala, Uganda. As a result, the design traditions which the local application design was based on were largely unestablished.

Within this research, two principal sources for path dependencies or generations were identified. The first path dependency had a social basis and guided the ideation and to some extent the design of the applications as it stem from the societal routines that were chosen to form the foundations for the applications. The second path dependency had a technological basis and originated from the technological resources used in the application building stage, which will be discussed in more detail in the following section (7.1.3). Regarding the first one, ideation and design, the existing societal routines provided a basis on top of which the applications were built, and in this sense these applications formed a continuum for the evolution of the societal routines as they tried to restructure and change them in some particular way. This seemed to be the norm in the application ideation and design stage, and although in some cases applications were more viewed as copies from other non-local applications, the general thinking among the developers and users was that these types of applications tended to fail.

Thus, the social path dependency occurred in relation to routines, where the intention of applications was to renew and restructure the performance of the chosen routines. In terms of application development, the questions related to design and path dependencies are not only about the paths that impact the application, but also (and to some extent more importantly) which path the application becomes a continuum of. Application development in a new location does not necessarily automatically fall into one particular path, but is more a result of a conscious choice between available paths. This can best be seen in the application development process when the start-ups choose a particular routine and aim to form a continuation for the routine's performance by changing it in some particular manner. In this sense it is crucial to understand that despite having inertia, the routines are not stable but keep on changing and evolving during their existence. As a result, the social path dependency a particular application gets linked to is based on choice,, i.e. choosing a certain routine over others, which is in line with Djelic et al.'s (2007) notion on path generation instead of dependency.

However, it is noteworthy that not every routine can form a basis for an application. In the selection process of routines, particular location-specific mechanisms impacted which routines could be targeted for application development and which not. These mechanisms functioned as filters for the existing routines, and it was down to the specifics of certain routines that decided the outcomes of the routine selection. As discussed in section 6.1.2, for example lack of available financial resources and investors meant that the chosen routines had to have some connection to generating revenue instantly. This could have been in the form of choosing a routine which performance required money changing hands at some stage of it, or then the routine reconstruction was seen as enabling to reach developmental goals, which NGOs or similar institutions were willing to support financially.

Overall, by making conscious choices on how to change the targeted routine the start-ups can have a say on the direction they would like the general societal development to head towards, which in practice is done by incorporating particular design principles into the applications (Oosterlaken, 2009, 2012). These design principles can also be advocated through the agendas of the technology hubs, as for some hubs there was the requirement for the start-ups to also address societal objectives in addition to the profit making. Through the process of restructuring existing routines, applications become agents of change which encapsulate certain values into their design (Latour, 1992; Winner, 1980). These notions of change become concretized in the attributes given to the different stages of routine performance as described in section 6.1.1.2. Borrowing Brown et al.'s (2010) distinction between the two types of ICT4D research (developing countries providing a particular context vs. the impact ICTs can have in developing countries), the process of attaching attributes into the different stages of routine performance represents a shift from viewing context as a state of how things are to one where there is an intention to shape that context in a particular way. Depending on the routines and attributes given to them, application development can be done according to the notions of appropriate design (Nieusma, 2004), feminist design (Rothschild, 1999) and ecological design (Ryn & Cowan, 1996) to name a few, but there is no guarantee as such that application development that takes place in developing countries would somehow automatically do so. Thus, application development has the potential of promoting particular types of change in societies, yet the form these changes take is ultimately dependent on the start-ups as well as the particular mechanisms affecting routine selection discussed in chapter 6. If there is an interest towards changing certain existing routines, this might require inputs from institutions such as local governments or NGOs to provide incentives for the start-ups to choose particular routines over others, and restructure them in ways that enable the applications to deliver the desired

outcomes. To some extent, this already occurs in the form of hackathons and tech hub agendas, yet especially in the case of the former the incentive tends to be relatively small and short-lived.

Application development provides developing countries possibilities to not only produce technologies but also to have the design process of the technologies happening within their borders. This has been seen of importance in the literature, since by being able to have design work done locally, these countries could have more influence within the global economy as they can come up with products that serve also their interests (Bonsiepe, 1991; Margolin, 2007).

However, as noted in section 3.2 of the conceptual framework chapter, the technological resources on top of which the applications are built have their origins outside the developing countries. As those resources impact significantly the shape the applications take, it is worthy to question how much of the design is truly produced locally and how much of it is still impacted by external influences, even though the start-ups do get to decide which technological resources to use. The role of technology gets highlighted especially during the actual building phase of the applications, where the developers use the technological resources available to them and turn those resources into applications.

7.1.3 Technological Resources as Building Blocks for Applications

The enthusiasm to develop software industries in developing countries can be partly linked to the availability of technological resources such as SDKs, APIs and different distribution platforms across the world. The digital materiality of these tools has meant that the process of developing applications has become more feasible everywhere as there is no need to physically transport the resources but instead those can be accessed at any location that has good enough connectivity to the internet. As noted earlier, the technological resources alone are not enough for applications to be created, as other aspects such as skills and hardware are also needed. However, in the stage of application development process where the actual application is built, these resources are of utmost importance as they provide the technological building blocks for the applications.

Although the developers have considerable room to choose and shape the technological resources, it can also be argued that in the end what the developers are doing is implementing resources originating from somewhere else. Many studies on technology implementation have tried to understand the reasons why technology implementation projects have tended to fail (e.g. Miscione 2007; James 2010; Bada 2002; Braa and Hedberg 2002). As noted in the section 2.4. of the literature review chapter, the importance of context of the implementation location has been repeatedly pointed out. In many cases the implemented technologies have not matched with the conditions posed by the local contexts, which has further affected the usage

and sustainability of these technologies (Walsham & Sahay, 2006). Different solutions have been proposed to the problem, most of which arguing for increased flexibility both in in relation to the technology and the receiving context (Bada, 2002; Macome, 2008; Njihia & Merali, 2013). Braa et al. (2002) for example proposed cultivation as a solution, which has the aim of linking the technology to the existing social systems over a period of time and this way secure the continued usage of the technology.

In relation to the results obtained from this research, the technological resources used in the application development process were viewed as quite agile as they enabled the developers to choose what type of applications to build and how, and overall the developers saw themselves able to shape the resources in different ways. Building the applications was often seen as relatively straight-forward, and the developers stated that the technological resources enabled them to create the applications they wanted, although in few occasions the technologies such as integrated payments were not available at the location at the time. Furthermore, the start-ups and the application developers were given support by different organisations in terms of their skills and other resources, and for example major technology companies such as Microsoft and Google were involved in providing training and organising competitions for the start-ups and giving funding for the winners. The presence of these companies was seen as mostly positive, partly because it sent a message that Kampala was viewed as a promising area for technology and software production. This is in line for example with the arguments put forward by Kshetri (2016), who noted that in addition to the technological resources these companies may be useful also in terms of offering the start-ups funding, resources, and training.

However, foreign technology companies and their local initiatives did not always make the matching of the local context to the used technologies easier. The events these companies were involved in organising and the trainings they promoted made in some regards the creation of successful applications more challenging. For example in the case of organizing competitions, participation to the competitions often implied that the applications had to be built onto the platform of the company, which in terms of the targeted users was not always the most optimal one. Furthermore, even though technological training is necessary to create more advanced technologies, it was not always clear whether the taught technologies were the most suitable ones for the application ideas of the start-ups. In this way, the presence of the major technological companies also had the impact of pushing the start-ups to wrong tracks in relation to the technological resources used for the applications.

Similarly, some areas remained problematic due to the origins of the companies providing the technological resources. As pointed out by the literature review, many of the technologies and

technological resources originate from companies that are based outside the developing regions in the Western countries (Caribou Digital, 2016). This has been seen as a form of knowledge imperialism, where the receiving locations are dependent of the knowledge of others (Ya'u, 2005). Many of the technologies used in application development can be claimed to been developed to serve issues relevant in the Western countries but less so of those of the developing countries (James, 2007; Singer, 1970). Regarding the results of this research, these issues were not seen explicitly stated as problems by the developers or start-ups. However, later on in the application development process questions rose on how well the chosen technological resources were able to provide solutions for the identified challenges of the targeted users.

In addition to the cases where the chosen technological resources were not necessarily the most appropriate ones, it also appeared that the technological solution models the start-ups used to the identified problems in the performance of local routines were not always ideal. These solution models did not stem from the technologies alone, but often came from the narratives built around famous technology start-ups like Uber and AirBnb on how they had solved particular problems. Therefore, the technological solutions that were offered for the identified challenges in existing routine performances took similar forms to those well-known technology companies, and in that way these technological narratives guided the application development process. Furthermore, because of the heavy focus on technology training for example by the technology hubs, what received often less attention was the actual problems and user needs and wants that the applications tried address. Overall, these factors meant often that there was a gap between the used technological solution models and the existing routines that provided the ideas for the applications: even though the ideas were based on a local context and technological resources had the built-in flexibility to shape the applications in different ways, there was tendency to use similar solution models that had been used elsewhere and which did not always fit the local context particularly well.

Other scholars have discussed the role of technology as a carrier of contexts, regulation, values and rationalisations (Latour, 1992; Lessig, 2006), or how technology simplifies real-life events (Kallinikos, Hasselbladh, et al., 2013) and promotes particular standards (Ciborra, 2005). All of these factors were present in the technological resources used for the applications, and also affected the application building and usage stages. However, this research argues that in addition to these, the technological narratives that came with the resources also affected how the applications were built. Instead of asking how to best solve the identified challenges in the chosen routine performances (assuming those had been identified correctly), the approach regarding what technologies to use and how to solve the challenges was often taken from other, non-local technology companies that had been successful globally. This phenomenon was

further intensified by the trainings provided by the technology hubs and global technology companies, which were valuable as such but gave less emphasis for studying the contextual factors first and only then seeing how appropriate the technological resources were to address the identified issues. The problems related to this type of approach surfaced in the usage stage of the applications, which in some cases also pegged the question on whether application development was the most suitable way to address the identified problems overall.

7.1.4 Usage of the Applications – Connecting Technology to Context

Following application design and building, the applications are launched to the targeted users with the hopes of these users using the applications on a frequent basis. Without usage the applications are literally of little use, and apart from the learning gained by the members of the start-ups the societal impact of the applications and the start-ups is likely to remain limited. As noted in the literature, innovations get their ultimate meaning by becoming part of social practices of their users (Tuomi, 2002), and if this does not occur, the applications remain largely meaningless from a societal perspective. Usage is also essential for developing the applications further, as through usage the start-ups can get feedback on whether the applications work well or in a less satisfactory manner (Iacob et al., 2013).

In addition to local technology production, technology consumption has also been seen as essential for developmental transformation to occur in the ICT driven areas of the economy (D'Costa, 2006). Among others, technologies can have a role in transferring skills to local communities and individuals (Dunn, 2009) but also in contributing to the development of the general technological infrastructure of societies, which enables the creation of businesses built around the new technological possibilities. For all of these to occur, consumption of the technologies is needed. However, the problem in developing countries especially in terms of more sophisticated technologies has often been that when consumption does occur, it is concentrated to the relatively few users from the more privileged segments of society (Parayil, 2005), as the wealthier social classes tend to have better access to more advanced devices and also are in possession of the skills to use the technologies.

However, based on the results of this research the consumption of the locally produced technologies was not guaranteed even among the people who had the devices and skills to use them. As discussed above, the differences in routine perceptions between the start-ups and the targeted users were often too big, which meant that the users saw less value in using the applications. In addition to this, the usage was also hindered by the users' perception of anything locally produced: if a product was locally made, for many that implied that it was of low quality and would fail to work. Local applications were therefore born with a certain type of stigma,

which meant that their position in the market was weaker from the start when compared to the applications made elsewhere. As a result, some of the start-ups were not too keen on highlighting their origins.

In relation to application usage among the less privileged segments of the society, some of the key factors that made it more difficult to target them were related to digital divide. Despite the fact that large part of the population had access to basic mobile phones and in this sense as indicated by Donner (2016) the divide regarding certain ICTs had narrowed, the more sophisticated technologies such as smartphones equipped with mobile data packages were still not common. This also had implications for the local application development. As discussed above, the campaigns made by technology companies such as Google and Microsoft to promote application development using their platforms and tools were often in contrast with the overall smartphone penetration in the country, which still remained relatively low. A more beneficial approach could have been to focus on application development for basic phones instead. Furthermore, following van Dijk's (2006) categorization of the different areas of digital divide, in many cases it was not clear if people saw any particular benefits in the local applications or even in the ownership of smartphones overall.

At the same time the general enthusiasm towards application development and especially around particular type of applications often directed the application developers and start-ups to choose technological resources that were not necessarily best suited to target the local market. As a result, even though the users had crossed the divide in relation to basic ICTs, many of the developers and start-ups had moved on to the next level of the divide and built applications for devices that were not yet that common. In quite a few cases this practically meant that the applications could not be used by many of the targeted users and the content the applications provided failed to reach the targeted users. In the literature one area of digital divide has been about content, especially regarding local content (Napoli & Karppinen, 2013; Souter, 2011). The somewhat misguided choices regarding technological resources meant that the applications often contributed little to closing the content divide by creating services to devices that were not widely used by a vast number of users. However, designing applications to one technological platform instead of another also comes with a cost, and for example creating applications for simple basic phones also meant that many of the functionalities available for smartphone applications were not there for the developers working with basic phones. In addition, from the perspective of the developers the choice of building smartphone applications was also linked to future projections about the growth rates in having access to smartphones and to other more sophisticated technologies.

In sum, targeting the more marginalised segments of the society was challenging because of the factors linked to digital divide. However, one particular area of application design appeared promising in this regard. Platforms, which concentrated on connecting two separate groups for example in terms of providing services, appeared to gain traction also among the poorer societal segments. In most cases these platforms enabled them to increase their incomes by providing more job opportunities. What is noteworthy though is that the companies providing the platforms equally struggled with the issues linked to digital divide. Many of the activities that were hoped to be taken care automatically had to be done manually, for example by calling instead of using the application. In other words, on an idea level the platform type of approach seemed to work relatively well, but the technological realisation of that idea did not fit with the realities of the targeted users.

This was not only because of lack of access to technologies, but also due to differences in perceptions or in the manners people had been accustomed to performing the routines in question. For example, it was not given that the presentation a map provides was similarly understood by the developer and the user. People often appeared to ultimately prefer the existing way of performing the routine. For example in relation to an application providing laundry services, it was sometimes the case that the application was only used to make the initial contact with a washer, after which the performance of the routine fell back to the existing form which did not include app usage. Therefore, the question on usage was not only about access to technologies or having the skills to use those, but also about how well the proposed technological approaches fit with the contextual factors and the societal routines that existed in the particular location.

Regarding software development, the argument has been made that it is dependent on the requirements set to it by societal factors (Fuggetta & Di Nitto, 2014). In a similar vein, continued usage requires the technologies to become first institutionalised into the societies (Kimaro & Nhampossa, 2005), and that for ICTs to be used in a meaningful way, the deeper societal structures need to be adapted as well (D'Costa, 2003). Based on the results from this research, it could be argued that the digital divide occurred also on the level of societal routines, and restructuring societal routines by using the technological resources to build applications was in many cases difficult. The existing routines had a certain type of inertia, which made them more difficult to change, but also the inability from the part of the start-ups to understand the routines as well as choosing less than ideal technological resources contributed to the failing of many of the applications. Therefore, digital divide should also be understand as context specific, and instead of only asking whether people have the necessary resources, skills and motivations to acquire and use the technologies, the question should be targeted more what is being hoped

from closing the divide regarding particular technology. Furthermore, technological resources should be chosen in a manner that they are compatible with the users' current devices and abilities. Situations and contexts do change over time, but there should be a balance between the current situation and what possibly lies ahead, and overall the start-ups should be better aligned with the technological realities that exist in their societies. This would enable closing the divides in other areas such as access to locally produced content.

Overall, the lack of technology in the existing routines also meant that developing applications which would have been somehow essential for the users to perform certain routines was relatively difficult. As most of the existing routine performances did not have an ICT component in them, there were limited amount of local technological resources to draw from, which also complicated the development of applications that could have provided more relevant content for the local users. In the existing setting, even if there was a technology incorporated into a routine performance, it was in most cases relatively isolated and as such could be easily bypassed. The argument could be made though that the existing application developers and technology start-ups in Kampala are pioneers, who are preparing the ground for other start-ups by giving incentives for users to acquire more advanced devices and also for device manufacturers to provide those, and as a result, enable the creation of applications that have more impact on societies than the local applications currently have.

7.1.5 The Connection between Application Development and Societal Development

Societal development is a concept of which multiple definitions exist on what is meant with development and what should be its objectives. These views on development range from theories like capability approach (Sen, 1999) to the more traditional ones such as modernization theory (Bernstein, 1971) and basic needs approach (Streeten, 1981). Despite of their somewhat different views on development, the common nominator to these different theories is that they all aim for a change for the better of the societies, although as noted the issue on what needs to be changed and made better is not entirely agreed upon. Overall, this notion of change is also the item that ties application development together with societal development. Even though the two are in many ways considered different for good reasons, they both thrive to change something in reference to the existing conditions.

Similar to societal development, application development can be seen as a process where existing societal routines are changed in some particular way according to the views of the start-ups creating the applications. As shown in the analysis, some of the different stages of routine performance are given attributes that indicate how they could be improved. At least part of this

improvement is expected to come from the functioning and usage of the applications, as they cover some if not all the stages of the routine performance. As a result, application development can be seen as a process of routine restructuration. This type of view provides a more holistic approach to application design, building and usage, as it covers also social aspects in addition to the technical ones.

Applications therefore represent intentions to make the routines stemming from the local context better, yet what is meant with better is dependent on the start-ups behind the applications. Each one of them may hold different views of what betterment means, and different applications might even be contradicting in their objectives. In addition, following Toyama's (2011) views it is not given that the applications would always be aiming for improvement of certain societal routines. The development of applications can be largely extrinsically motivated as it enables the start-ups to reach some other goal such as generating profit, and in that process any routine change, be that good or bad, which enables them to reach those goals will do. However, based on the data analysis done in chapter 6, most if not all of the start-ups did see their applications as improving the existing routine they had picked for their application, and even if this had not been the case, the start-ups aimed nevertheless to shape the society in some way by restructuring existing routines. As noted, the change was usually based on the application, although also the other stages of the routine performance had possible gone through changes in the routine restructuration process.

As a result, through their ostensive aspects the applications become scripts for the new, technologically mediated routines, which represented start-ups' views on how the existing routines should be made better. However, if this definition of betterment or change was not agreed upon by the targeted users, then usage was unlikely to follow. As a result, societal development based on changing existing routines was also subject to the people who performed the routine, and in relation to application development, the applications had to be able to provide the users something that they considered beneficial. Otherwise the applications remained ostensive without performative aspects, as the newly structured routines and the applications with them failed to get incorporated into the routine repertoires of the targeted users. The argument could be made that through usage, the applications that do aim for betterment of the society apply rather well to the Senian (Sen, 1999) notion on providing freedoms that people value, since if they don't, the applications most likely will not be used.

Regarding the literature on the impact of technology on the development of societies, the argument made here is that the success of technologies and application development overall is subject to the existing societal routines and their fit with the technology. This is also why the

results differ between locations in relation to the appropriation of same technologies. By understanding context in terms of societal routines it can be easier to pinpoint the reasons in a more detailed manner on why certain technologies have more success in some locations than others. Technology's ability to address developmental challenges can be ultimately tied to those societal routines. Even though ICTs are likely to contribute positively to the development of societies as pointed by ICT4D scholars (Duncan-Howell & Lee, 2008; Zainudeen & Ratnadiwakara, 2011), their impact could be further enhanced if more attention would be paid to the existing societal routines and understanding technologies' role through the existing routines.

Furthermore, instead of looking at the purely macro-level factors such as institutional settings, educational levels and general infrastructure (James, 2007; Kimaro & Nhampossa, 2005) to explain technology appropriation, analysis of existing routines provides an alternative perspective to the issues, and might also enable the researchers to have a deeper understanding on the kinds of technologies that are best suited to address the encountered societal challenges. For instance the finding on the greater impacts of mobile telephony in low-income countries (Swanson et al., 2006; Waverman et al., 2005) is a less of a surprise when understood how mobile telephony has been able to incorporate itself into the existing routines and therefore also provides a more fertile ground for other developmental benefits to occur.

What is noteworthy is that routine restructuration in itself does not imply the usage of any particular version of development theory. It merely argues is that whatever the objective is, it makes more sense to study the relevant routines first and come up with solutions to restructure the routines accordingly to reach the objectives. For example in relation to capability approach (Robeyns, 2005; Sen, 1999), the objective would be to understand how to restructure routines in a manner that enables people to increase their capabilities. Similarly, if development is understood in relation to the particular indicators, understanding first the societal routines that make it more difficult to reach certain levels of those indicators might provide better outcomes and also a different perspective to the ones of only addressing shortcoming in funding or lack of infrastructure. As an example of this, mobile phones' main implication for development has been linked to enabling access to information and increasing connectedness (Smith et al., 2011). By concentrating on the routines instead of the technologies, the question that arises is then what kind of information and connectivity is relevant for a particular routine to change.

In relation to Avgerou's (2010) note on the impacts of ICT applications to be either progressive (applications deliver substantial impacts by creating new income or capabilities, yet keep the prevailing deeper structures and mechanisms intact) or transforming (applications having more

fundamental impacts in terms of creating new business models and affecting societal power positions), the argument this research aims to make is that this depends on the societal routines the applications are ultimately based on. By successfully restructuring the routines, ICTs in their part can impact the value certain actors capture from performing those routines, and as a result, alter their overall position in the society. However, both progressive and transforming applications are based on the shifts and changes in routine performance, and on the perceptions that are linked to these routines. Even though ICTs may facilitate change, it is the routines and their restructuration that decides how certain routines become valued in the society, and how far-reaching impacts the restructuration of routines can have.

As a result, the intentions of many developing countries to nurture thriving software industries that in addition to economic benefits can help solving societal challenges appears sensible on a macro-level. At the same time, this type of development easily forgets the micro-level aspects by not asking how these technologies can be used to shape particular societal routines in a manner that would enable the economic benefits to follow. This is in line with Madon's (2004) argument that there is a need to shift from more technical measurements to understanding the actual impacts that technologies may have in their location of implementation.

This view also highlights the discussion on the role of technology in shaping societies. As seen in the analysis, the ideas for the applications stem from the local routines, but the solution models, at least in the case of applications, are based on the technological narratives and resources. As a consequence, the adoption of the applications can become more difficult as the technological solution models may not be appropriate for solving the local challenges, but it also entails that the successful cases of application development, where the applications are accepted and used by the targeted users, impose a particular technological functioning to the areas of the society where these applications are used.

As Walsham (2002) noted, there was a tendency to view ICT-enabled globalization as making cultures and societies more homogeneous. Cairncross (2001) for example argued that as a result of these developments societies would resemble each other to such a degree that the importance of context in technology or ICT usage would lose its meaning. Greig (2002) further claimed that eventually same technologies could be used in different locations with similar results, no matter the context or place. Other scholars have argued against this (e.g. Graham 2013; Wang, Lai, and Sui 2003) as discussed in section 2.1.2, pointing out for example that people's understanding of information and technology is highly context specific (Morgan, 2004), and how technologies go through a process of indigenization and adaptation in their location of appropriation (Robertson, 1992). Furthermore, different histories, geographies and languages

impact technology appropriation, and the diversity in values, customs and contexts cause differences in understandings and usages of similar technological artefacts and tools (Appadurai, 1996; Walsham, 2001).

On the basis of the analysis of this research, application development seems to lie between these two contrasting views on technologies' impact on societies. The ideas for the applications may differ considerably across locations, as they are often based on societal routines that may vary significantly when moving from one context to another. However, at the same time the ways applications aim to solve the challenges identified in the ideas tend to be similar. Example of this can be found from the data of this research, where areas like providing safer motorcycle taxis or having people come over to one's house to do laundry, both of which are relatively context-specific routines, where approached by using technological solution models similar to the ones used by companies like Uber or AirBnB. These solution models pose a particular way of functioning, such as using an app to order transportation or choosing from a large pool of service providers to get something done. Furthermore, as ICTs and their usage across the globe are likely to grow over time (Castells, 2011; Lee & Whitley, 2002), it also means that this similarity in technological solution models across locations will increase unless there are more distinct solution models available.

A whole other question is how much of an issue this is, since in order for a technological solution model to succeed, it has to be accepted by the users especially in cases where they are free to choose whether to use the applications or not. This also implies that if applications are used, there is a fair possibility that the users themselves see these technologies useful. However, in case there is a need to shift away from the solution models imposed by the prevailing technologies, then local technology production that is less dependent on external technologies should be promoted. As noted, although the development of the applications is done locally, it still relies on technological resources that have their origins elsewhere. If those resources were to be local as well, it might also imply greater heterogeneity in the solution models posed by the technologies.

7.2 Implications of the Research

Based on the discussion of the results of this research, the methodological, theoretical and practical implications stemming from the research are presented below.

7.2.1 Methodological Implications

The methodological implications of this research are twofold. The first one relates to the observed relativity of the key critical realist principles, which impact the process of choosing the

suitable data sources in research using critical realist approach. Secondly, seeing routines as proxies to context provides a methodological tool to conceptualize and operationalise context.

7.2.1.1 Relativism in Critical Realist Research

When identifying the relevant mechanisms for understanding the role of context in application development, relativity surfaced in actors' abilities to perceive these mechanisms and also the two other critical realist notions of events and the empirical. As Klein (2004) stated, "*the real are the causal mechanisms and structures that produce actual events a subset of which then is empirically observed*" (p. 131). In relation to critical realist research, the causal mechanisms and structures as well as events are viewed as largely hidden. However, what constituted as a causal mechanism and especially events and the empirical seemed to be relative to the situations and to the actors' position in those situations.

For the users of the applications, a critical realist event can be claimed to be the application itself and the empirical the application's interfaces which deliver the information for the user to perceive it. Thus, a mechanism for a user is the whole process of the application development, in which an idea is turned into an actual application and distributed into the user's device. By looking at the interface alone, the user is able to make assumptions on what kinds of mechanisms or which sort of reality lead to the application to be developed and distributed. However, as these are largely hidden, the only empirical the user observes is the application user interface and the instructions that come with the application. From those she or he can then try to infer which kinds of events and mechanisms have been at play to produce something like the application, but little else. The assumptions related to the applications are slightly to be more accurate if the user has some understanding of any of the areas related to application development, or if she or he has somehow been involved in the process of developing the application for example by testing the earlier versions of the applications, but even then the empirical remains as relatively limited to the user.

The actors who are closer to the application development process have a broader view of the empirical than the user. As a result, for the developers and other persons involved in the work the mechanisms, events and the empirically observed turn out to be different in comparison to the users. For example an application developer has often a complete understanding of the code that creates the interfaces for the users. Thus, what is an event for the user becomes the empirical for the developer. Similarly, the developer or some other person in the start-up might have a relatively clear understanding of the areas that for the user display as mechanisms, which cannot really be observed but have to be inferred from the empirical. Mechanism for the developer or for the start-up might again be areas such as what led someone to create a certain

programming language or how the design of the hardware that the developer uses was decided upon.

In research, the identification of mechanisms that explain certain events and observed outcomes is often done from the perspective of the researcher, who collects data that enables her or him to infer what kinds of mechanisms could explain the observed outcomes. If we accept the actor-based relativity in relation to the key critical realist concepts, one could argue that in the role of the researcher is not so much about deducing what kinds of mechanisms might be in play but especially in the beginning of the research process more one of finding the actors and data sources that provide the best view of the mechanisms. It can even be asked if there are any mechanisms that are completely hidden from all in the sense that critical realism argues in relation to the real. However, even the actors and data that have the best view of the studied mechanism are unlikely to provide completely objective accounts of those mechanisms. This again emphasizes the researcher's role as an assessor on what mechanisms are more likely to be the closest ones to the real, and assuming she or he has done this correctly.

What this means though is that even though generative mechanisms provide a useful tool for analysing certain events and outcomes, the researcher should first make sure that the mechanisms that she or he seeks to unveil are truly mechanisms in the critical realist sense for large enough audiences, so that time is not spent on trying to explain something that is only a mechanism for the researcher while completely clear for the other relevant actors. Especially in research on topics like application development which includes different actors such as developers, investors and targeted users, the perspective each of them possesses varies considerably. For the researcher, the mechanisms that need to be studied are those that combine aspects that are relevant for each of the actors, and the contribution that can be made using generative mechanisms is one that provides a holistic view of the research object that also includes the interactions that occur between these actors.

In sum, the critical realist notions of mechanisms, events and the empirical display themselves differently to different actors, and as a result, there is amount of relativity in what is considered for example a mechanism in a particular situation and by whom. Although this is of less importance within the confines of this research, it stresses the importance of choosing one's data sources for the research. This research combined different sources, i.e. documents, fieldnotes and interviews from the start-ups' personnel, people relevant to the field and targeted users, and the information sought from a particular source was determined by their position, as the intention was to find sources that had the closest view to the mechanisms that were researched. As the main interest of this research was in the application development

process and the role of context in it, the interviews of start-ups and their personnel provided the main bulk of data, although other relevant actors, such as the targeted users, were also interviewed to highlight the aspects they had good understanding of such as usage. Overall, for research a using critical realist approach, the relativity in mechanisms, events and the empirical is worthy to take into account, especially when choosing the data sources for the research.

7.2.1.2 Routines as Proxies to Context

In addition to the relativity in the key concepts of critical realism, another methodological implication of this research is the use of routines as proxies to context. Routines can be used as a tool to conceptualise and operationalize context, and for example in a research that aims to use contextual variables, routines may provide a useful source of those variables. As routines tend to be based on the contexts that are closest to them, they also transmit the changes that occur in the general context. Overall, how routines are performed and also which types of routines exist in a particular location tell something about the norms, values and beliefs that are in place.

In some cases it might be necessary to go deeper to understand the origins of those contextual factors, but even then a researcher can start from the existing routines and move forward by asking questions on why a certain routine is performed in a particular way. In this sense, studying routines first can provide a relatively concrete way to understand issues related to context and people's perceptions of it. This does not mean that other sources of context research should be discarded, but more that also those sources can be linked to the performance of the routines and that routines can form a lens through which context is understood.

Although simplified, seeing situations and events as a set of routines can offer a useful tool to research areas such as action research. Research looking at existing routines may enable to better capture contextual changes, and by viewing how routines and their performance have changed over time it can also be easier to link those changes to larger societal changes. In other words, routines can provide a bottom-up type of approach instead of first looking at macro-level events and infer their meaning top-down to the level of individual actors.

7.2.2 Theoretical Implications

The key theoretical implications of this research are connected to the notion of viewing application development as a routine restructuration, in which technological narratives often have a role in dictating the solution models proposed by the applications. In addition, by seeing application development as a routine-based process it can be linked together conceptually with societal development.

7.2.2.1 Application Development as Routine Restructuration

The main theoretical implication of this research is to view application development as a process of routine restructuration, where chosen routines form the basis of the applications and in the process these routines get restructured and changed in a manner that involves the usage of an application. What is noteworthy is that the technological stage of building the application forms only one part of this routine restructuration. As a result, this type of view enables a more holistic understanding of the application development, and moves the focus away from a purely technological construction to something that incorporates social dimensions into the process. It further stresses the point on technology having a supporting role in the routine restructuration, and even this role can be questioned if it appears that a particular technology is not the best way forward to reach the wanted outcomes.

Furthermore, a routine-based view of application development enables us to see application development process as rooted into the surrounding socio-cultural environment and linked to the routines that people carry out performing in particular locations. It also displays a way how in the design stage certain value systems and worldviews can be incorporated into the design of artefacts. This occurs firstly by choosing the routines that are targeted and secondly by thinking how to change those chosen routines. As noted, this is not to say that changing routines would be somehow deterministic or easy, as unwanted routine changes may occur as well or in case the applications are not used, the routine restructuration has no impact. However, routine restructuration enables the relevant actors to create a roadmap, which can be further evaluated by others for example in terms of meeting the expected outcomes.

As noted, no routine change will occur if the applications fail to gain traction among the targeted users. Some of the reasons impeding adoption in a developing country context are linked to the digital divides that exist within the society. Even though there is a large part of the population that has access to some ICTs such as basic mobile phones (Donner, 2016; Gitau et al., 2010), the technology start-ups themselves often position them on a more sophisticated technological level by designing applications which only function in more advanced devices such as smartphones. This highlights the dynamic nature of digital divide, where one divide might be closed but new ones open as technologies develop. Overall, there exists a certain paradox as many these more sophisticated technologies are actively promoted for application development in Kampala but at the same time those technologies might not be the best ones to reach large numbers of users in the area or in the country. As a result, many start-ups do not stay in the level where the divide has been largely closed but have moved on to the next one. However, in order to reach a larger number of users, they have to accommodate the users who are not in possession of the needed sophisticated technologies by offering alternative means to use their

service without using the application, for example by calling or texting. This again means more manual work for the start-ups as they need resources to take care of these added functionalities.

In relation to the technology and the technological resources used for the applications, research has shown how technologies not only enable but also control what can and cannot be done (section 3.2). This is also true in application development, where the technological resources impose the ultimate limits on what kinds of applications can be created and how they need to function. However, what this research has aimed to show is that it is not only the technological resources but also the narratives that come with the technologies that affect how applications get developed. Although the basis of the applications stems from the local context and the routines that exist in it, the solution models, or the ways that the locally created applications aim to address the identified challenges in the existing routines, are conditioned by the success stories of global technology companies as these solutions are often copied to the local applications as well. In reference to the debate on the effect of ICTs to the world, this research argues that if those solution models are accepted by the targeted users, certain routine performances may end up resembling each other across different locations, even though the basis for performing those routines can be quite different originally.

However, the technological solution models were often unable to address the challenges identified in local routines and as a result, the targeted users appeared to prefer the existing routines over the proposed ones. This is not uncommon overall, as many existing routines have certain inertia which makes changing them more difficult. However, it also appeared that many of the chosen routines themselves were not really ideal for providing technological solutions. As a result, this research would like to argue that in addition to access, skills and motivations, digital divide can also be found on the level of routines and in their incompatibility to be technologized in the form of applications. In other words, the inability of many applications to attract usage went in some cases beyond the more typical issues discussed in the literature on digital divide like access to technologies and skills to use these technologies. The issue seemed more that there was a divide also on the level of societal routines, where particular routines appeared difficult to be captured by using technologies, and there existed a difference between various routines in terms of their ability to form a basis for the applications. To some extent this was linked to the existing filtering mechanisms (discussed in 6.1.2) that made certain routines unattractive for the developers to build their applications on. However, also the low level of technicity in the performance of those routines was an issue as it made it more difficult to build applications around those routines.

By using the concepts of code and space (Kitchin & Dodge, 2011), the digital divide in terms of routines can be seen as instances where code and space do not meet and therefore it is also more unlikely for transductions where code and space are in a dyadic relation to occur. This is partly due to the lack of technologies in the space, which further complicates the development of applications that would be essential in performing certain routines or transductions. As noted, also the transductions themselves impact this, and some transductions may be more open for technological solutions than others. Therefore when discussing the relation between code and space, this research suggests that there is a need to also understand the particular factors that make spaces more accessible for code and allow coded spaces or code/space to occur. Furthermore, it might also make sense to add to the categories of code/space, coded space and background spaces a fourth category of *codeless space* to highlight the circumstances in which the spaces with code may get created. Naturally, as ICTs are increasingly used it might be just a matter of time when code becomes mixed with space in all the different locations of the globe. However, as this research has aimed to show, the issue requires a more detailed understanding of routines themselves and how they enable or constrain the incorporation of code into them.

From a developing country perspective, the problem for the developers is also the low levels of existing technicity. In practice this means that when the developers are restructuring a particular routine by building an application around it, it is likely that only few of the stages of routine performance can be technologized in any way. If there was more existing technicity available in routine performances, it would be easier for the developers to automate additional stages of the routine performance. This would also increase the chances of the applications forming part of code/space or coded spaces and gaining more popularity among the users. In many ways, this is also what sets application development occurring in developing countries apart from the more developed countries. Even though many of the technological resources can be acquired elsewhere, there are very few local resources available that would enable the development of applications that could incorporate and automate several stages of routine performance into them. By doing so, these local technological resources would also allow the companies to reduce the number of tasks requiring manual labour, making it easier for them to scale up, among other things.

7.2.2.2 Societal Development as a Process of Routine Change and Adoption

In the literature many studies on ICTs and development highlight the need to take contextual factors better into account. This pattern occurs across different areas of ICT4D literature such as technology design, technology implementation and technology impact research. In these studies the relevant contextual factors are often highlighted and their relation to the technologies in question analysed. The findings provide valuable information of the key factors

that are relevant for technology appropriation in the locations and also enable the readers to understand how context and technology might be interlinked in the studied cases.

However, although the importance of context has been recognised in the literature, the question on what constitute as relevant contextual factors in one particular case is not always straightforward to analyse or applicable to another. This research proposes that instead of looking context as a whole, a useful starting point can be to study the routines that are in place in a particular location, and through these routines make inferences about the larger context of the location. Furthermore, approaching context through routines also enables the researcher to analyse the ways the identified routines need to be changed in order to achieve developmental objectives. Similar to application developers that aim to change particular stages of routine performances, research with a developmental objective can look which type of routine changes would enable the research to meet its goals. By understanding where a routine goes 'wrong' and giving attributes to that stage accordingly, it can also be easier to see the solutions for changing that routine and the routine's performance.

This is not to say in any means that the other approaches for researching development can be forgotten. The point is merely that routine-based development studies offer an additional perspective, which also provides a way of comparing different locations and evaluating the applicability of solutions developed elsewhere to new locations and contexts. To some extent, this has already been done in research on context, but making this in a more explicit manner may lead to further insights on the role of context in relation to technology and societal development. Although admitted that seeing societies as a sum of routines and routine performances is a simplification and does not capture every aspect of societal complexity, routines do provide a way to understand development in which context and its manifestations can be compared across different locations.

Overall, the model presented in figure 6.4 (section 6.1.2.3) provides a particular way of understanding application development as a process that originally stems from the societal routines that are found from the contexts and locations of application developers or the persons who first came up with the idea. The model also shows how certain mechanisms have a role to play in what type of routines are chosen in particular locations and how technology fits into the equation. Furthermore, it connects application development into broader contextual changes that occur through the usage of the created applications, and does this while allowing explanations on how application development differ between locations of different technological base levels. As such, it provides a useful tool to conceptualize and theorize

application development as a socio-technical process placing special emphasis on the challenges that developers and other relevant factors are likely to encounter during this process.

7.2.3 Practical and Policy Implications

The application development process overall was seen as a beneficial experience by the start-ups even in cases where the applications failed to attract users. The members of the start-ups described having learned different skills in the process and saw that the experiences they had gained from starting a business of their own was likely to impact their future careers positively. Furthermore, some of the start-ups did achieve varying levels of success, and in some cases the benefits also trickled down to the poorer members of the society in the form of increased incomes. However, in many of these cases the applications themselves did not necessarily play a major role, but it was more the idea and the routine restructuration behind the business that was seen as addressing the needs and wants of the targeted users.

Many of the more promising start-ups had spent considerable time first studying the routines they were aiming to restructure in some particular way. However, most of the start-ups went ahead technology first, without really understanding the realities they were trying to change with the application. Although in some cases the shortcomings of this approach were possible to fix after receiving feedback from users, it also occurred that the inability to understand the targeted routines well enough led the start-ups to close their businesses. Overall, the enthusiasm around technology is easy to understand for example by looking at the type of trainings that were offered in the technology hubs or the kinds of events that were organised. Although not negating the importance of having good technical skills, it appeared that the technology hubs often organised training mainly on how to use particular technologies, and in some cases those technologies were not even the most apt ones to reach the majority of the users in the region. In addition, many of the hackathons that were organised were built around developing applications or other technologies yet the challenges these applications were trying solve were not understood well enough.

In other words, instead of studying and evaluating the challenges the applications were trying to address, there was often a tendency to move relatively quickly to the application building stage and using technological resources that from the perspective of the targeted users made little sense. What this implies is that instead of providing training on different technologies or even on business management, the technology hubs could place more emphasis on providing training on problem definition and different solution models, some of which would not even have to be technological. Even in the case that the context changes rapidly and within few years most of the targeted do have smartphones, it would still be essential to understand well the

challenges the applications are trying to solve before building the actual applications. As a result, this research suggests that the technology hubs and organizations organizing events like hackathons would take a step away from the technological approach and concentrate more on studying and understanding the issues the start-ups are trying to address. This would have the effect of not only increasing the number of successful start-ups, but also making sure that the start-ups could provide solutions that are truly useful for the targeted users. The solutions could have a technological form, but this would have to be decided on the basis on what works best for the users and not as automatically given.

Studying the targeted routine first is also important because as shown above, being present in particular location did not guarantee knowing the routines automatically. Even though it was more likely that the local start-ups were better aware of the local context, this did not mount to knowing perfectly every local routine. Therefore, being present in a particular location is a necessary but not a sufficient condition for understanding local realities well enough, and applies not only to application development but also to any project related to technology appropriation. As a result, even though the technology start-ups have the potential to impact the socio-economic conditions in a particular location and supporting them financially and otherwise is most likely beneficial, on a policy level it might be a good idea not to give in for all the enthusiasm around technologies and instead concentrate on the issues first and only then consider technologies' role as parts of the solutions.

7.3 Further Research

On the basis of the results and theoretical implications, there are few areas for further research that could be useful to understand better application development occurring in developing countries and also elsewhere. For most part they are linked to the notion of context and its relation to technology. However, additional research on the values and contexts that technology carries with it, be that in the form of the technological resources or narratives, could enhance our understanding of technology production in particular locations.

7.3.1 Improving the Routine-Based Model on Application Development

The model developed in this research is based on the findings from application developers and technology start-ups in Kampala, Uganda. As noted in the analysis, the key mechanisms depend on the type of start-ups that are being researched and some of the identified mechanisms only apply to start-ups based in Kampala or in similar locations. As an example, the mechanism regarding funding is likely to be quite different in locations where there are more start-up investors. In addition, a higher level of abstraction especially regarding the identified societal and technological filters might make the model more applicable to these other locations.

Furthermore, although the model describes quite well the application development process in Kampala, it does not mean that it would be complete or perfect. Thus, any additional research that uses the model is likely to improve it, and this also applies to cases in which the research is about topics other than application development. For example, even though this research does give some indication on the mechanisms affecting routine selection, more research could go into the reasons why application developers choose particular routines over others in different locations, and as a result provide a more abstract categorization of these reasons. Another area in the model that could be more detailed is the one concerning application development and the role of technological resources and narratives in it.

7.3.2 Technology as a Carrier of Contexts

In the literature it is widely recognised that technology functions as a carrier of context for example in the form of standards, but more research could be done in terms of the values that are incorporated into technologies and also on the technological narratives that guide the development of technologies such as applications. As discussed in this research, the technological narratives provided examples on how to use technological resources to solve identified challenges in the chosen routines. It is unclear though how common this approach is, and what are the key drivers behind it. In general, research focusing on the values and norms incorporated into technologies is important as ICTs and other technologies have more pervasive roles in the everyday lives of people in different locations. Furthermore, the technological narratives have a more global reach as a result of increasing global connectivity.

Overall, this type of discussion is already taking place for example in the field of artificial intelligence regarding its ethical dimensions. However, another aspect to this discussion would be the kinds of structures that are being transported with the spread of technologies to new locations. One suggestion regarding this is for example using the McDonaldization theory by Ritzer (Ritzer, 2007), which argues that due to globalization, values such as efficiency, calculability, predictability and control are making their way to different areas of societies across the globe. Assuming that this view is correct, the question that follows is how much of it is due to the increasing usage of different technologies, and even in the case that Ritzer's view is not accurate, the question on the type of changes which occur in societies due to their increasing reliance on particular type of technologies remains valid. Overall, answering this question would help in understanding the types of societies technologies enable to build, and also how far local contextual factors can impact the process.

7.3.3 Technology Cultures

In contrast to the research area discussed above on technology as a carrier of contexts, local technology production may also lead to the birth of new technology cultures. As this research has argued, the applications have their basis in the existing societal routines that are being restructured in the application development process. Although the technological resources have an important role in dictating the shapes the applications take, there exist a variety of these resources and the start-ups can choose between them. Since the starting point for the applications are local routines, which are based on the existing context, it is plausible that the applications that stem from particular contexts are in some ways unique and differ from the ones developed in other contexts.

As a result, in theory it is possible that different regions and contexts produce different kinds of applications and also technologies as whole. This would again mean that there exist different technology cultures, in which every region or context produces particular kinds of technologies, in a manner that is spoken of different food cultures. This kind of development may further become intensified as more regions become technology producers themselves. Even though application development still relies very much on technological resources that mostly have their origins outside the developing countries, if these countries become producers of technological resources it is likely to further contribute to the creation of local technology cultures.

As noted, this kind of development is also being countered by the global reach of many of the existing technological resources, as they can be acquired over the Internet no matter the location, which may lead to greater resemblances between applications created in different locations. Furthermore, technology production is not the result of the existing context alone and as a process is quite complex capturing many different factors. However, from a research perspective it would be interesting to understand how for example applications created in one location differ from others built in another location and context. Research on technology cultures would further contribute to our knowledge on the role of context in technology production.

7.4 Limitations

7.4.1 Generalisability

As often in case study research, generalisability of the results can be questioned. Although in this research the start-up scene in Kampala can in many ways seen as a typical case of application development occurring in a developing country context that poses its own set of characteristics to the process, it is also clear that each location is unique in some aspects and there are clear contextual differences between locations. As noted in relation to critical realist research, same

technologies may not activate same mechanisms in different places (Klecun, Lichtner, Cornford, & Petrakaki, 2014), and it is difficult to make claims of validity across these different locations. As a result, it can be that the mechanisms identified in this research differ from the mechanisms that exist elsewhere, as also pointed out in the section on areas for future research (section 7.3). However, even if this is the case it does not rule out of the possibility that existing routines form the basis of application developed in other locations or that technological resources impact the process in particular ways, yet in order to verify this further research is needed.

The above relates also to the scope and focus of this research. The research concentrated on application development that took place in a developing country context, and many of the findings were linked to the particular contextual factors that exist in this kind of environment. However, it is important to remember the large variety of countries that fall under the term developing country. There are a lot of differences between them, and not least in terms of the steps taken in innovation activities and technological advancements. As a result, even though the routine-based model on application water is likely to largely apply in several locations, it is clear that for example in relation to tradition in software development, the thickness of the local technological base that developers can draw resources from or access to smartphones and other technological tools among targeted users can be quite different in places like China, Brazil and Uganda. Therefore the results speak mostly to countries that are on a similar developmental stage as Uganda or Kampala to be more specific.

In a similar manner, it is not entirely clear how well the results of this research apply to the production of other technologies. Application development differs in many ways from other technology production areas such as hardware design. Therefore it is noteworthy to use the routine-based model outside application development with caution, especially if those technologies offer varying degrees of adaptability or are less reliant on external technological resources. Furthermore, start-ups themselves vary in size and operation models, which also influence the technology development process. In this case all the start-ups were relatively small, consisting at most of approximately 15 members. In addition, for all of them, the first target market was the local one. As a result, companies that are larger and target different markets in addition to their own may have approaches that vary significantly from the ones studied in this research.

Despite of these limitations on generalisability, the research hopes that the findings and the models based on the conceptual framework and data analysis is applicable also in other locations and research contexts.

7.4.2 Data Collection

Another limitation of this research is linked to the data collection. As a first point, interviews always possess a certain amount of subjectivity, and especially when interpreting the meanings there lies a risk that those interpretations do not match with the ones meant by the interviewees. To avoid this from happening, multiple data collection methods were used such as non-participant observation and documents to triangulate the findings from the interviews. Furthermore, the interviews were analysed in a manner that did not leave more room than necessary for the researchers' interpretations, as the statements made by interviewees were often taken as is without questioning too much their truthfulness or other intentions behind the answers.

However, related to this and another limitation linked to data collection was the positioning of the researcher during the fieldtrips. It often turned out that due to his ethnicity (white European) the researcher was sometimes seen as a possible investor by the interviewees. Extra care was taken to explain the interviewees the reasons for the interviews and also by indicating that the researcher was not going to invest into the start-ups. In most cases, it appeared that this worked relatively well, however in few occasions there were some doubts about how the interviewees viewed the researcher and whether that affected their answers. Therefore, the claims from those interviews were placed under additional scrutiny by the researcher and also more effort was put to use additional sources of data to verify those claims. What also helped in these situations was the ways interviews were performed and what kind of data was looked for. Regarding the first, there was an intention by the researcher to create a relaxed atmosphere for the interviews for example by first having informal chats about mundane events before starting with the actual interview. In relation to the data that was sought from the interviews, it was not the aim of the researcher to get the interviewees to reveal confidential information, but more to just talk about their experiences regarding the topics of the interviews. Despite of these efforts, there is a risk that answers given to the researcher were not always entirely accurate.

7.4.3 Conceptual

The usage of routines as proxies to context provides a relatively clear approach for studying contextual factors that are relevant in particular locations and settings. However, this approach might not always be capable of covering all the richness of context, as context can also be understood for example from a more systemic level. Therefore, for research that focuses on context it is worthy to consider the most suitable way to understand context and define context in manner that best serves the purposes of that research. Overall, understanding context through societal routines has provided a useful lens for this thesis, but at the same time it cannot be entirely guaranteed that routines capture all the relevant aspects of context.

7.5 Conclusion

The focus of this research has been on understanding the contextual dimensions of application development that occurs in developing countries. The motivation for this was the increase in device ownership and Internet connectivity across the globe, which in general has been met with great enthusiasm as many developing countries have been eager to reap the possible benefits following these developments. One of the areas in this regard has been application development, and in many Sub-Saharan African countries there have been initiatives to promote the development of application development ecosystems. Countries like Kenya, South Africa and Nigeria have led the process, but also other countries such as Uganda have aimed to follow suit.

These initiatives and the discussion around the topic have been largely technology driven, and large parts of the literature have been aimed at identifying the key variables that need to be met in order for an application development sector to thrive. Less research however has gone into understanding the impacts of local contexts and even in cases where this has been done, the topic has mainly been approached from the macro-level. Although these studies are important as such and have shed further light to different aspects of application development, there has been very little research that has studied the topic from the perspective of the key actors, such as application developers and technology start-ups, which is what this research has aimed to do.

The research was done by using qualitative methods and a critical realist approach. It aimed to identify mechanisms that produced the observed outcomes, i.e. the empirical, and took the form a case study by studying application development occurring in Kampala, Uganda. One of the methodological implications that stem from the research was the relativity of the critical realist notions of mechanisms, events and the empirical, which appeared to vary according to the actor in question. What displayed itself as a mechanism for one actor was for another the empirical, depending on the position this actor occupied. Therefore when using critical realist research approach, it is of use to see the position of a particular data source in relation to the phenomenon that is under research.

The principal research objectives were the following: first, it tried to understand how the context, where applications are created and hopefully used, affected the application development process for example in terms of the types of applications that are developed. Second, the research looked into the role of technology in the application development process, which is of interest especially since many of the technological resources needed in application development came from outside the location studied. Finally, both of these two areas were studied from a more micro-level perspective, in which the intention was to understand the topic

from the perspective of the start-ups, key institutions and targeted users, instead of only looking at key policies or actions by global corporations that were related to the topic.

As this research has aimed to show, application development for local markets should be seen as a routine restructuration process, where a particular routine is chosen to form the basis for the applications. The routine and its performance are then divided into different stages, which are given attributes to indicate how those stages should be changed. Some of these changes, if not all, are incorporated into an application that is developed to somehow improve the performance of that routine in terms of addressing needs and wants that have been identified among the targeted users. After building the applications, it is launched to the targeted users with the hopes that they will adopt the application and with it incorporate the new restructured routine into their routine repertoire.

However, there are several challenges in developing the applications and getting the users to adopt the applications. The first one relates to the ability of the start-ups to perceive the chosen routines and their performances correctly. Sometimes the ways the start-ups viewed these routines differed considerably from the perceptions of the targeted users. Furthermore, not every routine could be chosen due to social and technological mechanisms that filtered out some of the routines. Finally, the technological solution models that came with the technological resources and narratives also made it sometimes difficult for the start-ups to attract large numbers of users.

Overall, the start-ups and the institutions around them, such as technology hubs, seemed often to neglect the need to have a correct understanding of the chosen routines. Many start-ups started the application development process by building the applications first, which meant that the application development became largely a technological process and less attention was given to the relevant social factors. In relation to the technology, the solution models for the identified challenges that the applications aimed to solve were often copied from the technological narratives that came with the technological resources. These technological narratives were basically success stories on start-ups such as Uber or AirBnb, and the technological solutions for the locally developed applications basically resembled these applications that had been success elsewhere. In this sense, although the basis of the applications was in local routines, the technological solution models were borrowed from somewhere else. What was noticeable was that the start-ups which first studied the chosen routines extensively and only then moved on to build the applications appeared to do better in terms of attracting users and usage. However, in an environment where many existing routines are performed largely without relying on technologies, it was often difficult to create

applications that would enable the users to do something that they could not do without the applications.

In sum, this research makes the argument of seeing application development as a routine restructuration process, where existing routines are actively shaped by the start-ups on the basis of the perceptions the start-ups have of those routines. The research stresses the need to see technologies as part of routine performances, and also in relation to areas of literature such as digital divide, it argues for focusing more on the targeted routines instead of solely looking at the technologies in terms of people's access or skills. This type of routine-based view of application development further suggests that societal development can be seen as a series of changes in existing routines and in the performances of those routines. What is noteworthy is that within this routine restructuration, material characteristics may have a role but do not dictate the process. As a matter of fact, changing the routines and their performance in a meaningful way does not necessarily require any technologies at all, especially if the technologies do not seem to match with the routines in question or with the desired change.

8 Appendix

A: List of Interviews

Id	Interviewee	Length(s)	Mode/ Location(s)	Date(s)
1	Entrepreneur, developer, application now developed in Nairobi, Kenya (Application A)	1h 15 min	Café, London	08/10/2014
2	Developer, contracted by a start-up, Nairobi, Kenya (Application A, Skype interview)	38 minutes	Skype	04/11/2014
3	Developer, start-up partner, based in Kampala, Uganda (Application B + C, 3 interviews in total)	1h 12 min, 38 minutes, 51 minutes	Hive Colab, Kampala (all three interviews)	06/07/2015, 19/11/2015, 24/06/2016
4	Developer, start-up partner, based in Kampala, Uganda (Application B)	27 minutes	Hive Colab, Kampala	08/07/2015
5	Developer, start-up partner, based in Kampala, Uganda (Application B)	25 minutes	Hive Colab, Kampala	09/07/2015
6	Developer, start-up partner, based in Kampala, Uganda (Application B + C, 2 interviews in total)	40 minutes, 30 minutes	Bar, Kampala; Hive Colab, Kampala	10/07/2015, 08/12/2015
7	Developer, start-up owner, based in Kampala, Uganda (Application D)	55 minutes	Hive Colab, Kampala	11/07/2015
8	Developer, start-up owner, based in Kampala, Uganda (Application E, 2 interviews in total)	49 minutes, 1h 10min	Hive Colab, Kampala; Bar, Kampala	09/07/2015, 26/07/2016
9	Developer, start-up owner, based in Kampala, Uganda (Application E)	29 minutes	Hive Colab, Kampala	06/07/2016
10	Owner and director of a start-up, Kampala, Uganda (Application F, 2 interviews in total)	1h 6 min, 58 minutes	Hive Colab, Kampala	13/07/2015, 12/07/2016
11	Developer, start-up partner, based in Kampala, Uganda (Application G)	57 minutes	Office, Kampala	11/12/2015
12	Developer, start-up partner, based in Kampala, Uganda (Application G)	44 minutes	Office, Kampala	11/12/2015
13	Owner and manager of a start-up, Kigali, Ruanda (Application H)	25 minutes	Hotel, Nairobi	22/07/2015
14	Developer, start-up partner, Kampala, Uganda (Application I)	34 minutes	Hive Colab, Kampala	10/07/2015
15	Game designer, start-up owner, Kampala (Application J, 3 interviews in total)	32 minutes, 41 minutes, 1 h 11 min	Hive Colab, Kampala; Outbox, Kampala; Café, Kampala	14/07/2015, 01/12/2015, 29/07/2016

16	Developer, Start-up partner, University lecturer, Kampala (Application K)	26 minutes	Makerere University, Kampala	21/07/2016
17	Developer, Start-up partner, University student, Kampala (Application K)	20 minutes	Pulse Lab, Kampala	28/06/2016
18	Developer, Start-up partner, University researcher, Kampala (Application L)	1h 10 min	Makerere University, Kampala	11/07/2016
19	Owner and director of a start-up, Kampala, Uganda (Application M)	1h 2 min	Hive Colab, Kampala	05/07/2015
20	Owner and director of a start-up, Kampala, Uganda (Application N)	59 minutes	Café, Kampala	19/07/2016
21	Developer, start-up owner, Kampala, Uganda (Application O)	44 minutes	Hive Colab, Kampala	05/07/2015
22	IT company owner and developer, Kampala, Uganda (Developed several applications)	1h 8min	Office, Kampala	22/07/2016
23	Developer, start-up partner, Kampala, Uganda (Application P + Hardware design)	37 minutes	Outbox, Kampala	03/12/2015
24	Owner and director of a start-up, Kampala, Uganda (Application Q)	42 minutes	Office, Kampala	03/12/2015
25	Developer, start-up partner, Kampala, Uganda (Application R)	29 minutes	Hive Colab, Kampala	06/07/2016
26	Developer, start-up partner, Kampala, Uganda (Application R)	32 minutes, 41 minutes, 1 h 20 min	Hive Colab, Kampala	09/07/2016
27	Developer, start-up partner, Kampala, Uganda (Application S)	31 minutes	Office, Kampala	21/07/2016
28	Designer, start-up partner, Kampala, Uganda (Application T)	48 minutes	Skype	17/11/2016
29	Developer, start-up partner, Kampala, Uganda (Application U)	30 minutes	Hive Colab, Kampala	18/07/2016
30	Entrepreneur, start-up owner, Kampala, Uganda (Hardware)	43 minutes	Office, Kampala	10/12/2015
31	Start-up owner and manager, Kampala, Uganda (Application V)	41 minutes	Outbox, Kampala	08/12/2015
32	Designer, start-up partner, Kampala, Uganda (Web development)	1h 16 min	Hotel, Kampala	23/07/2016
33	Developer, start-up partner, Kampala, Uganda (Web development for local companies, 2 interviews in total)	23 minutes, 36 minutes	Hive Colab, Kampala; Café, Kampala	08/07/2015, 04/07/2016
34	Developer, working for local companies, Kampala, Uganda (Different kinds of applications developed)	1h 4min	Office, Kampala	02/08/2016

35	Developer, start-up partner, Kampala, Uganda (Web development for local companies)	36 minutes	Outbox, Kampala	19/07/2016
36	Technology Hub Research Manager, Nairobi, Kenya	28 minutes	iHub, Kenya	20/07/2015
37	Country director for Uganda of an IT company that provides communication solutions such as bulk SMS, short codes and premium SMS, USSD, and MMS, Kampala	16 minutes	Hive Colab, Kampala	07/07/2015
38	Innovation Lab Director, Kampala	25 minutes	Hive Colab, Kampala	11/07/2016
39	Technology Hub Communications Manager, Kampala	16 minutes	Hive Colab, Kampala	29/06/2016
40	Technology Hub Director, Kampala	24 minutes	Outbox, Kampala	04/12/2016
41	Technology Hub Vice Director, Kampala	31 minutes	Outbox, Kampala	11/12/2015
42	Mobile Operator Country Director, Kampala	23 minutes	Bar, Kampala	09/07/2015
43	University lecturer, innovation lab advisor, Kampala	20 minutes	Pulse Lab, Kampala	28/06/2016
44	Technology Blogger, Kampala	1h 11 min	Café, Kampala	12/07/2016
45	Foreign diplomat in Uganda, responsible of ICT co-operation	47 minutes	Office, Kampala	08/07/2015
46	University Professor of ICT, Kampala	26 minutes	Makerere University, Kampala	14/07/2016
47	User of application B	12 minutes	Café, Kampala	05/12/2015
48	User of application B	11 minutes	Café, Kampala	08/07/2016
49	User of application B	12 minutes	Café, Kampala	22/07/2016
50	User of application F	9 minutes	Motorcycle Taxi, Kampala	01/12/2015
51	User of application F	10 minutes	Motorcycle Taxi, Kampala	07/12/2015
52	User of application F	11 minutes	Motorcycle Taxi, Kampala	09/07/2016
53	User of application F	9 minutes	Motorcycle taxi assembly point, Kampala	22/07/2016
54	Part of target group of application F	20 minutes	Bar, Kampala	01/08/2016
55	Part of target group of application F	12 minutes	Café, Kampala	02/08/2016
56	User of application R	11 minutes	Office, Kampala	15/07/2016
57	User of application R	10 minutes	Café, Kampala	15/07/2016

58	User of application developed in Uganda (not listed above)	12 minutes	Hotel, Kampala	11/07/2015
59	User of application developed in Uganda (not listed above)	9 minutes	Café, Kampala	12/07/2015
60	User of application developed in Uganda (not listed above)	14 minutes	Café, Kampala	12/07/2015
61	User of application developed in Uganda (not listed above)	12 minutes	Café, Kampala	27/07/2016
62	User of application developed in Uganda (not listed above)	18 minutes	Hotel, Kampala	29/07/2016
63	User of application developed in Uganda (not listed above)	15 minutes	Bar, Kampala	29/07/2016
64	User of application developed in Uganda (not listed above)	10 minutes	Café, Kampala	03/08/2016

B: Fieldnotes

Fieldnotes 1-10 gathered during the three stays in the field (each fieldnote approximately from a period of two weeks)

July 2015

Fieldnotes 1: Fieldnotes on observations done in the tech hubs in Kampala

Fieldnotes 2: Fieldnotes on observations about Kampala and Nairobi and tech hub in Kampala and Nairobi

November-December 2015

Fieldnotes 3: Fieldnotes on observations done in the tech hubs and events organised in Kampala

Fieldnotes 4: Fieldnotes on observations done in the tech hubs in Kampala

Fieldnotes 5: Fieldnotes on observations done in the tech hubs and in a pitching event in Kampala

Fieldnotes 6: Fieldnotes on observations done in the tech hubs in Kampala

June-August 2017

Fieldnotes 7: Fieldnotes on observations done in the tech hubs in Kampala

Fieldnotes 8: Fieldnotes on observations done in the tech hubs and workshops in Kampala

Fieldnotes 9: Fieldnotes on observations done in the tech hubs, in the office of a technology start-up and weekly seminars at the local university in Kampala

Fieldnotes 10: Fieldnotes on observations done in the tech hubs and in hackathons/pitching events in Kampala

C: Documents

Document 1: Screenshot from a Google Play store of application B

Document 2: Screenshot from a Facebook post of a tech hub

Document 3: Screenshot from a Google Play store of application F

Document 4: Flyer of application F

Document 5: Screenshot from a Facebook post of a tech hub

Document 6: Poster of application K

Document 7: Screenshot from a website of application N

Document 8: Screenshot from a Facebook post of a tech hub

Document 9: Screenshot from a Google Play store of application H

Document 10: Screenshot from a Google Play store of application F

Document 11: Screenshot from a Facebook post of a tech hub

Document 12: Screenshot from a Google Play store of application B

Document 13: Screenshot from a Facebook post of a tech hub

Document 14: Screenshot from a website of application U

Document 15: Screenshot from a Google Play store of application V

Document 16: Screenshot from a Facebook post of a tech hub

Document 17: Screenshot from a Facebook post of a tech hub

Document 18: Screenshot from a Google Play store of application R

Document 19: Screenshot from a Google Play store of application F

D: Examples of the type of applications encountered in the field

Type of application	Description	Stage (at the time of fieldwork)	Operating system	Letter in Appendix A and C (if interviewed)
App for selling and buying agricultural products	Algorithm linking buyers with sellers (farmers) of agricultural products that runs on basic phones	Launched	SMS-based	L
Motorcycle taxi application	Enables users to find a motorcycle taxi of the company, which are known to be safer than the other motorcycle taxis. Current version enables mobile money payments.	Launched	Android, iOS	F
Laundry service application	A platform linking laundry washers with people who need their laundry washed, based on subscription fees at the time of the data collection, downloads at around 1000s in Google Play store.	Launched	Android	B
Coupon payment system	Instead of cash, provides coupons for specific purposes that can be given to persons and thus restrict the use of money to certain purpose.	Design phase	Android, USSD	D
NGO-management tool	A management tool for NGOs to take care of collaboration, accounting, HR and various other aspects of NGO activities	Trial phase	Web-based	R
Application for harvest management for farmers	Tool that provides farmers relevant information and provides access to services like harvest-stocking harvest.	Launched	Web, USSD, SMS, Android, iOS	O
A school management tool	A tool that takes care of the different areas of school management such as accounting and communication between parents and teachers and enables parents to track student performance.	Trial phase	Desktop application, Android	I
App for providing trustworthy garages	Platform linking car owners to trustworthy garages, aims to provide also a quality guarantee of the garages	Trial phase	Android	G
Application for sharing study material and notes	Application to share study material across students going to different schools	Trial phase	Desktop application, Android	M
Software for providing various payment systems for customers	Enables merchants to accept payments of various kinds, including mobile money and credit cards, operates across various countries.	Launched	API	N
Tool for hospital management	Takes care of basic managerial and other operations in hospitals such as accounting and storage	Launched	Desktop application	S
Portable doppler pregnancy device	Doppler connected to a smartphone	Trial phase	Windows mobile	Q
Motorcycle taxi application	Enables users to find a motorcycle taxi of the company, which are known to be safer than the other motorcycle taxis. Current version enables mobile money payments and uses sensor data to estimate driver safeness (operates in Kigali, Rwanda).	Launched	Android, iOS	H
Mobile money application	Enables users to receive mobile money payments into their bank accounts	Launched	Android	C

Agriculture platform	Enables selling and buying agricultural products as well as provides collaboration between different stakeholders (farmers, NGOs, traders)	Trial phase	Android	V
Job portal for developers	Platform for connecting Ugandan and other nationality developers for IT projects abroad	Launched	Web-based	
Online store solution	Provided the online store functionality for stores (relied on existing platform solutions)	Launched	Web-based	U
Point of sales tool for restaurants	Restaurant point of sales software that also functions as a storage tracker	Trial phase	Desktop application	E
Portable tool for testing HPV infection	Hardware device connected to a smartphone that enables the users to test from their urine whether they have an HPV infection and also provides support for contacting medical care and tracks testing	Trial phase	Windows mobile	P
Game with the intention to increase HIV awareness	A game that asks the player questions related to HIV while playing in order to raise HIV awareness among youth	Trial phase	Android	J
Tool for organising SMS surveys	An application that enables the sending and collection of SMS based surveys	Launched	Desktop application, SMS	A
Career planning platform	A web-site that connects students to career advice services and guides them in their course choices	Building phase	Web	R
Tool for plant disease diagnosis	A mobile application that enables to check whether a cassava plant is suffering from particular diseases, also uploading the information to a central database.	Launched	Windows mobile, Linux	K
App for playing an Ugandan card game	One of the most famous applications developed in Uganda, enables the users to play a particular card game in their mobiles, more than half a million downloads in the Google Play store.	Launched	Android, iOS	
Tourism app	An application providing information from tourist attractions located in Uganda	Launched	Android, iOS	
Address app	Provides each location a code that can be used as an address	Building phase	Android	T

9 References

Abbott, P. (2013). How can African Countries Advance their Outsourcing Industries: An overview of possible approaches. *The African Journal of Information Systems*, *5*, 27–35.

Abbott, P., & Jones, M. (2012). Everywhere and nowhere: nearshore software development in the context of globalisation. *European Journal of Information Systems*, *21*, 529–551.

Adams, P. C., & Ghose, R. (2003). India.com: the construction of a space between. *Progress in Human Geography*, *27*, 414–437.

African Development Bank. (2016). *African Development Report 2015 - Growth, Poverty and Inequality Nexus: Overcoming Barriers to Sustainable Development*. Retrieved from https://www.afdb.org/en/documents/document/african-development-report-2015-growth-poverty-and-inequality-nexus-overcoming-barriers-to-sustainable-development-89715/

Ahrne, G. (1994). *Social Organizations: Interaction Inside, Outside and Between Organizations*. London, UK: SAGE Publications.

Akrich, M. (1992). The De-Scription of Technical Objects. In W. E. Bijker & J. Law (Eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change*. London, UK: MIT Press.

Al-Jaghoub, S., & Westrup, C. (2009). Reassessing Social Inclusion and Digital Divides. *Journal of Information, Communication and Ethics in Society*, *7*, 146–158.

Ananny, M., & Winters, N. (2007). Designing for Development: Understanding One Laptop Per Child in Its Historical Context. In *2007 International Conference on Information and Communication Technologies and Development* (pp. 1–12).

Antonelli, C. (2003). The digital divide: understanding the economics of new information and communication technology in the global economy. *Information Economics and Policy*, *15*, 173–199.

Appadurai, A. (1996). *Modernity at Large: Cultural Dimensions of Globalization*. Minneapolis, USA: University of Minnesota Press.

Archer, M. (1995). *Realist Social Theory: the Morphogenetic Approach.* Cambridge, UK: Cambridge University Press.

Attride-Stirling, J. (2001). Thematic Networks: an Analytic Tool for Qualitative Research. *Qualitative Research*, *1*, 385–405.

Avgerou, C. (2001). The Significance of Context in Information Systems and Organizational Change. *Information Systems Journal*, *11*, 43–63.

Avgerou, C. (2008). Information Systems in Developing Countries: A Critical Research Review. *Journal of Information Technology*, *23*, 133–146.

Avgerou, C. (2010). Discourses on ICT and Development. *Information Technologies & International Development*, *6*, 1–18.

Avgerou, C., Hayes, N., & Rovere, R. L. L. (2016). Growth in ICT Uptake in Developing Countries: New Users, New Uses, New Challenges. *Journal of Information Technology*, *31*, 329–333.

Avgerou, C., & Madon, S. (2004). Framing IS Studies: Understanding the Social Context of IS Innovation. In F. Land, C. Avgerou, & C. Ciborra (Eds.), *The Social Study of Information and Communication Technology : Innovation, Actors, and Contexts* (pp. 162–182). Oxford, UK: Oxford University Press.

Avgerou, C., & Walsham, G. (2000). Introduction: IT in Developing Countries. In C. Avgerou & G. Walsham (Eds.), *Information Technology in Context: Studies from the Perspective of Developing Countries*. London,UK: Ashgate Publishing Company.

Bada, A. O. (2002). Local Adaptations to Global Trends: A Study of an IT-Based Organizational Change Program in a Nigerian Bank. *The Information Society*, *18*, 77–86.

Bailey, D. E., Leonardi, P. M., & Barley, S. R. (2011). The Lure of the Virtual. *Organization Science*, *23*, 1485–1504.

Barad, K. (2003). Posthumanist Performativity: Toward an Understanding of How Matter Comes to Matter. *Signs*, *28*, 801–831.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, *11*, 369–386.

Benkler, Y. (2006). *The Wealth of Networks : How Social Production Transforms Markets and Freedom*. New Haven, USA: Yale University Press.

Bernstein, H. (1971). Modernization Theory and the Sociological Study of Development. *The Journal of Development Studies*, *7*, 141–160.

Bhaskar, R. (2008). *A Realist Theory of Science* (2nd ed.). New York, United States: Routledge.

Bijker, W. E. (1995). *Of Bicycles, Bakelites, and Bulbs: Toward a Theory of Sociotechnical Change*. Massachusetts, USA: MIT Press.

Blumer, H. (1980). Mead and Blumer: The Convergent Methodological Perspectives of Social Behaviorism and Symbolic Interactionism. *American Sociological Review*, *45*, 409–419.

Bonsiepe, G. (1991). Developing Countries: Awareness of Design and the Peripheral Condition. In C. Pirovano (Ed.), *History of Industrial Design: 1919–1990 the Dominion of Design*. Milan, Italy: Electa.

Borgmann, A. (2010). Reality And Technology. *Cambridge Journal of Economics*, *34*, 27–35.

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

Braa, J., & Hedberg, C. (2002). The Struggle for District-Based Health Information Systems in South Africa. *The Information Society*, *18*, 113–127.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*, 77–101.

Bright, J., & Hruby, A. (2015). The Rise Of Silicon Savannah And Africa's Tech Movement. Retrieved from http://social.techcrunch.com/2015/07/23/the-rise-of-silicon-savannah-and-africas-tech-movement/

Britz, J. J., Lor, P. J., Coetzee, I. E. M., & Bester, B. C. (2006). Africa As a Knowledge Society. *International Information & Library Review*, *38*, 25–40.

Brown, A. E., & Grant, G. G. (2010). Highlighting the Duality of the ICT and Development Research Agenda. *Information Technology for Development*, *16*, 96–111.

Byrne, E., Nicholson, B., & Salem, F. (2011). Information Communication Technologies and the Millennium Development Goals. *Inf. Technol. Dev.*, *17*, 1–3.

Cairncross, F. (2001). *The Death of Distance: How the Communications Revolution is Changing Our Lives*. Harvard Business Press.

Caribou Digital. (2016). *Winners and Losers in the Global App Economy*. Farnham, Surrey, United Kingdom: Caribou Digital Publishing. Retrieved from http://cariboudigital.net/winners-and-losers-in-the-global-app-economy/

Carmel, E. (2003). The New Software Exporting Nations: Impacts on National Well Being Resulting from Their Software Exporting Industries. *The Electronic Journal of Information Systems in Developing Countries*, *13*, 1–6.

Carmel, E., Lacity, M. C., & Doty, A. (2014). The Impact of Impact Sourcing: Framing a Research Agenda. In R. Hirschheim, A. Heinzl, & J. Dibbern (Eds.), *Information Systems Outsourcing* (pp. 397–429). Berlin, Germany: Springer.

Castells, M. (2011). *The Rise of the Network Society: The Information Age: Economy, Society, and Culture*. West Sussex, UK: John Wiley & Sons.

Chavula, H. K. (2013). Telecommunications development and economic growth in Africa. *Information Technology for Development*, *19*, 5–23.

Checkland, P., & Scholes, J. (1999). *Soft systems methodology: a 30-year retrospective*. Chichester, UK: Wiley.

Chen, G., & Kotz, D. (2000). *A Survey of Context-Aware Mobile Computing Research* (Technical Report TR2000-381 No. Vol. 1. No. 2.1.). Hanover, NH, USA: Dartmouth College.

Cherkasky, T. (2004). Design Style: Changing Dominant Design Practice. *Design Issues*, *20*, 25–39.

Chigona, W., Beukes, D., Vally, J., & Tanner, M. (2009). Can Mobile Internet Help Alleviate Social Exclusion in Developing Countries? *The Electronic Journal of Information Systems in Developing Countries*, *36*. Retrieved from https://www.ejisdc.org/ojs2.../index.php/ejisdc/article/view/535

Chircu, A. M., & Mahajan, V. (2009). PERSPECTIVE: Revisiting the Digital Divide: An Analysis of Mobile Technology Depth and Service Breadth in the BRIC Countries. *Journal of Product Innovation Management*, *26*, 455–466.

Ciborra. (2005). Interpreting e-Government and Development: Efficiency, Transparency or Governance at a Distance? *Information Technology & People*, *18*, 260–279.

Clifford, J., & Marcus, G. E. (1986). *Writing Culture: The Poetics and Politics of Ethnography*. Berkeley, United States: University of California Press.

Cohen, M. D., & Bacdayan, P. (1994). Organizational Routines Are Stored as Procedural Memory: Evidence from a Laboratory Study. *Organization Science*, *5*, 554–568.

Cohen, M. D., Burkhart, R., Dosi, G., Egidi, M., Marengo, L., Warglien, M., & Winter, S. (1996). Routines and Other Recurring Action Patterns of Organizations: Contemporary Research Issues. *Industrial and Corporate Change*, *5*, 653–698.

Cohendet, P. S., & Simon, L. O. (2016). Always Playable: Recombining Routines for Creative Efficiency at Ubisoft Montreal's Video Game Studio. *Organization Science*, *27*, 614–632.

Cooper, R., & Law, J. (1995). Organization: Distal and proximal views. In S. B. Bacharach, P. Gagliardi, & B. Mundell (Eds.), *Research in the Sociology of Organizations: Studies of Organizations in the European Tradition* (pp. 275–301).

Corral, L., Sillitti, A., & Succi, G. (2014). Software assurance practices for mobile applications. *Computing*, 1–22.

Crotty, M. (2003). *The Foundations of Social Research : Meaning and Perspective in the Research Process*. London, UK: Sage.

Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems Journal*, *8*, 273–289.

David, P. A. (1985). Clio and the Economics of QWERTY. *The American Economic Review*, *75*, 332–337.

David, P. A. (2001). Path Dependence, Its Critics and the Quest for "Historical Economics." In P. Garrouste, S. Ioannides, & E. A. for E. P. Economy (Eds.), *Evolution and Path Dependence in Economic Ideas: Past and Present* (pp. 15–40). Cheltenham, UK: Edward Elgar Publishing.

D'Costa, A. P. (2003). Uneven and Combined Development: Understanding India's Software Exports. *World Development*, *31*, 211–226.

D'Costa, A. P. (2006). Introduction: Charting a New Development Trajectory? In *The New Economy in Development* (pp. 1–13). Palgrave Macmillan, London.

Dearden, A., & Haider Rizvi, S. M. (2015). ICT4D and Participatory Design. In *The International Encyclopedia of Digital Communication and Society*. John Wiley & Sons, Inc. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/9781118767771.wbiedcs131/abstract

Deloitte. (2016). *Uganda Economic Outlook 2016*. Retrieved from https://www2.deloitte.com/content/dam/Deloitte/ug/Documents/tax/Economic%20Outlook%202016%20UG.pdf

Desai, P., Purohit, D., Stadje, W., & Wong, P.-K. (1998). Leveraging the Global Information Revolution for Economic Development: Singapore's Evolving Information Industry Strategy. *Information Systems Research*, *9*, 323–341.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications. *Human-Computer Interaction*, *16*, 97–166.

Díaz Andrade, A., & Urquhart, C. (2010). The affordances of actor network theory in ICT for development research. *Information Technology & People*, *23*, 352–374.

Dilley, R. M. (2002). The problem of context in social and cultural anthropology. *Language & Communication*, *22*, 437–456.

Djelic, M.-L., & Quack, S. (2007). Overcoming path dependency: path generation in open systems. *Theory and Society*, *36*, 161–186.

D'Mello, M. (2005). "Thinking Local, Acting Global": Issues of Identity and Related Tensions in Global Software Organizations in India. *The Electronic Journal of Information Systems in Developing Countries*, *22*. Retrieved from http://www.ejisdc.org/ojs2/index.php/ejisdc/article/view/164

Dodge, M., & Kitchin, R. (2004). Flying through Code/Space: The Real Virtuality of Air Travel. *Environment and Planning A: Economy and Space*, *36*, 195–211.

Dodge, M., & Kitchin, R. (2005). Code and the Transduction of Space. *Annals of the Association of American Geographers*, *95*, 162–180.

Donner, J. (2016). *After Access: Inclusion, Development, and a More Mobile Internet* (1 edition). Cambridge, Massachusetts, USA: MIT Press.

Donner, J., & Escobari, M. X. (2010). A Review of Evidence on Mobile Use by Micro and Small Enterprises in Developing Countries. *Journal of International Development*, *22*, 641–658.

Dosi, G. (1982). Technological Paradigms and Technological Trajectories: A Suggested Interpretation of the Determinants and Directions of Technical Change. *Research Policy*, *11*, 147–162.

Duncan-Howell, J. A., & Lee, K.-T. (2008). Policy Catch Up: Developing Nations and Developing ICT Policy Documents. In *Faculty of Education*. Herriot-Watt University, Edinburgh, UK: British Educational Research Association (BERA). Retrieved from https://eprints.qut.edu.au/13651/

Dunn, H. (2009). Teleworking the Mobile Caribbean: Emerging Patterns of Broadband-Assisted Remote Work Among the Marginalized in Jamaica and Trinidad and Tobago. *Information Technologies & International Development*, *5*, 52–66.

Easton, G. (2010). Critical realism in case study research. *Industrial Marketing Management*, *39*, 118–128.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., & Yoo, Y. (2015). Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System. *MIS Quarterly*, *39*, 217–243.

Ekbia, H. R. (2009). Digital Artifacts as Quasi-Objects: Qualification, Mediation, and Materiality. *Journal of the American Society for Information Science and Technology*, *60*, 2554–2566.

Esterberg, K. G. (2002). *Qualitative Methods in Social Research*. McGraw-Hill Higher Education.

Evans, D. (2014). The technology ecosystem in Kampala, Uganda. Retrieved July 27, 2017, from https://vc4a.com/blog/2014/05/28/the-technology-ecosystem-in-kampala-uganda/

Evans, D. S., Hagiu, A., & Schmalensee, R. (2008). *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. Massachusetts, USA: MIT Press.

Ewing, J., Okolloh, O., & Rawlings, L. (2011). *Transforming the East African ICT Sector by Creating a Business Engine for SMEs*. Washington DC, USA: Wold Bank. Retrieved from http://www.infodev.org/articles/transforming-east-african-ict-sector-creating-business-engine-smes

Farrell, J., & Weiser, P. (2003). Modularity, Vertical Integration, and Open Access Policies: Towards a Convergence of Antitrust and Regulation in the Internet Age. *Harvard Journal of Law and Technology*, *17*, 85–134.

Faulkner, P., & Runde, J. (2012). On Sociomateriality. In P. M. Leonardi, B. A. Nardi, & J. Kallinikos (Eds.), *Materiality and Organizing* (pp. 49–66). Oxford University Press.

Faulkner, P., & Runde, J. (2013). Technological Objects, Social Positions, and the Transformational Model of Social Activity. *MIS Quarterly*, *37*, 803–818.

Feenberg, A. (1999). *Questioning Technology*. London, UK: Routledge.

Feldman, M. S. (2000). Organizational Routines as a Source of Continuous Change. *Organization Science*, *11*, 611–629.

Feldman, M. S., & Pentland, B. T. (2003). Reconceptualizing Organizational Routines as a Source of Flexibility and Change. *Administrative Science Quarterly*, *48*, 94–118.

Feldman, M. S., Pentland, B. T., D'Adderio, L., & Lazaric, N. (2016). Beyond Routines as Things: Introduction to the Special Issue on Routine Dynamics. *Organization Science*, *27*, 505–513.

Feldman, M. S., & Rafaeli, A. (2002). Organizational Routines as Sources of Connections and Understandings. *Journal of Management Studies*, *39*, 309–331.

Flick, U. (2009). *An Introduction to Qualitative Research*. London, UK: SAGE.

Flyvbjerg, B. (2006). Five Misunderstandings About Case-Study Research. *Qualitative Inquiry*, *12*, 219–245.

Fong, M. W. L. (2009). Technology Leapfrogging for Developing Countries. *Encyclopedia of Information Science and Technology, Second Edition*, 3707–3713.

Fontana, E., & Sorensen, C. (2005). From Idea to Blah! Understanding Mobile Services Development as Interactive Innovation. *Journal of Information Systems and Technology Management*, *2*, 101–120.

Foray, D. (1997). The Dynamic Implications of Increasing Returns: Technological Change and Path Dependent Inefficiency. *International Journal of Industrial Organization*, *15*, 733–752.

Forman, C. (2005). The Corporate Digital Divide: Determinants of Internet Adoption. *Management Science*, *51*, 641–654.

Foster, C., & Azmeh, S. (2016a). Digital Latecomer Economies and National Internet Policy: The Case of China. Presented at the The Internet, Policy and Politics, Oxford. Retrieved from http://ipp.oii.ox.ac.uk/2016/programme-2016/track-a-politics/government-iii-digital-government-and/christopher-foster-shamel-azmeh-digital

Foster, C., & Azmeh, S. (2016b). *The TPP and the Digital Trade Agenda: Digital Industrial Policy and Silicon Valley's Influence on New Trade Agreements* (LSE International Development Working Paper 175). Retrieved from http://www.lse.ac.uk/internationalDevelopment/publications/Working-Papers/WP175-ShamelAzmeh.aspx

Freeman, P. A. (2007). Is "designing" cyberinfrastructure -or even, defining it- possible? *First Monday*, *12*. Retrieved from http://firstmonday.org/ojs/index.php/fm/article/view/1900

Friederici, N., Ojanperä, S., & Graham, M. (2017). The Impact of Connectivity in Africa: Grand Visions and the Mirage of Inclusive Digital Development. *Electronic Journal of Information Systems in Developing Countries*. Retrieved from https://papers.ssrn.com/abstract=2855398

Friedman, T. L. (2007). *The World Is Flat: A Brief History of the Twenty-First Century*. London, UK: Penguin Books.

Fuchs, G., & Shapira, P. (2005). *Rethinking Regional Innovation and Change: Path Dependency or Regional Breakthrough*. Springer Science & Business Media.

Fuggetta, A., & Di Nitto, E. (2014). Software Process. In *Proceedings of the on Future of Software Engineering* (pp. 1–12). New York, NY, USA: ACM.

Garcia, L., & Quek, F. (1997). Qualitative Research in Information Systems: Time to be Subjective? In *Information Systems and Qualitative Research* (pp. 444–465). Springer, Boston, MA.

Garrido, M., Badshah, A., & Coward, C. (2009). Deconstructing ICT Skills and Employability. *Information Technologies & International Development*, *5*, iii–vii.

Gathigi, G., & Waititu, E. (2012). Coding for Development in the Silicon Savannah: The Emerging Role of Digital Technology in Kenya. In *Re-imagining Development Communication in Africa*. Lexington, USA: Lexington Books.

Gaver, W. W. (1991). Technology Affordances. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 79–84). New York, NY, USA: ACM.

Ghazawneh, A., & Henfridsson, O. (2013). Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model. *Information Systems Journal*, *23*, 173–192.

Gibson, J. J. (1977). The Theory of Affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving, Acting, and Knowing* (pp. 67–82). New York, USA: Lawrence Erlbaum Associates.

Gitau, S., Marsden, G., & Donner, J. (2010). After Access: Challenges Facing Mobile-only Internet Users in the Developing World. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2603–2606). New York, NY, USA: ACM.

Global Entrepreneurship Monitor. (2015). *GEM 2014 Global Report*. Retrieved from http://www.gemconsortium.org/report/49079

Goldsmith, B. (2014). The Smartphone App Economy and App Ecosystems. In G. Goggin & L. Hjorth (Eds.), *The Routledge Companion to Mobile Media*. Abingdon, England: Routledge (Taylor & Francis Group). Retrieved from http://www.routledge.com/books/details/9780415809474/

Government of Uganda. (2017). The Economy. Retrieved July 27, 2017, from /about-uganda/sector/economy

Graham, M. (2008). Warped Geographies of Development: The Internet and Theories of Economic Development. *Geography Compass*, *2*, 771–789.

Graham, M. (2013). Geography/Internet: Ethereal Alternate Dimensions of Cyberspace or Grounded Augmented Realities? *The Geographical Journal*, *179*, 177–182.

Graham, M., & Mann, L. (2013). Imagining a Silicon Savannah? Technological and Conceptual Connectivity in Kenya's BPO and Software Development Sectors. *The Electronic Journal*

of Information Systems in Developing Countries, *56*. Retrieved from http://144.214.55.140/Ojs2/index.php/ejisdc/article/view/1107

Green, N. (2002). On the Move: Technology, Mobility, and the Mediation of Social Time and Space. *The Information Society*, *18*, 281–292.

Greig, J. M. (2002). The End of Geography?: Globalization, Communications, and Culture in the International System. *The Journal of Conflict Resolution*, *46*, 225–243.

Grudin, J. (2001). Desituating Action: Digital Representation of Context. *Human-Computer Interaction*, *16*, 269–286.

GSMA. (2017). GSMA Mobile Economy 2017. Retrieved June 28, 2017, from https://www.gsma.com/mobileeconomy/

Gunkel, D. J. (2003). Second Thoughts: Toward a Critique of the Digital Divide. *New Media & Society*, *5*, 499–522.

Gurumurthy, A., & Singh, P. J. (2005). Political economy of the information society – A Southern view. *Information Society for the South: Vision or Hallucination*, 103–116.

Hacking, I. (1999). *The Social Construction of What?* Boston, USA: Harvard University Press.

Hammersley, M. (2003). Conversation Analysis and Discourse Analysis: Methods or Paradigms? *Discourse & Society*, *14*, 751–781.

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

Hargittai, E. (2002). Second-Level Digital Divide: Differences in People's Online Skills. *First Monday*, *7*. Retrieved from http://firstmonday.org/ojs/index.php/fm/article/view/942

Hargittai, E., & Walejko, G. (2008). The Participation Divide: Content Creation and Sharing in the Digital Age. *Information, Communication & Society*, *11*, 239–256.

Harindranath, G., & Sein, M. K. (2007). Revisiting the Role of ICT in Development. In *Proceedings of the IFIP TC 9*. Sao Paulo : IFIP WG9.4. Retrieved from https://pure.royalholloway.ac.uk/portal/en/publications/revisiting-the-role-of-ict-in-development(b8894704-75ff-4cff-9b8a-9cb0d32ea0cd).html

Hayes, N., & Westrup, C. (2012). Context and the processes of ICT for development. *Information and Organization*, *22*, 23–36.

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

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

Heeks, R. (2006). Using Competitive Advantage Theory to Analyze IT Sectors in Developing Countries: A Software Industry Case Analysis. *Information Technologies & International Development*, *3*, 5–34.

Heeks, R. (2008). ICT4D 2.0: The Next Phase of Applying ICT for International Development. *Computer*, *41*, 26–33.

Heeks, R. (2010). Do Information and Communication Technologies (ICTs) Contribute to Development? *Journal of International Development*, *22*, 625–640.

Heeks, R., & Molla, A. (2009). Impact Assessment of ICT-for-Development Projects: A Compendium of Approaches. Retrieved from https://www.research.manchester.ac.uk/portal/en/publications/impact-assessment-of-ictfordevelopment-projects-a-compendium-of-approaches(7ca09357-9c40-40be-9bce-f16dd97b9c46).html

Heim, K. (2009, June 29). Grameen Foundation and Google create mobile apps for Africa. *The Seattle Times*. Retrieved from http://www.seattletimes.com/business/grameen-foundation-and-google-create-mobile-apps-for-africa/

Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2012). Evaluating Cross-Platform Development Approaches for Mobile Applications. In *Web Information Systems and Technologies* (pp. 120–138). Berlin, Germany: Springer.

Henfridsson, O., Yoo, Y., & Svahn, F. (2009). Path Creation in Digital Innovation : A Multi-Layered Dialectics Perspective. *Association for Information Systems*, *9*. Retrieved from http://www.diva-portal.org/smash/record.jsf?pid=diva2:228420

Hirschheim, R. (1985). Information Systems Epistemology: An Historical Perspective. In *R. Galliers (Ed.), Information Systems Research: Issues, Methods and Practical Guidelines* (pp. 28–60). London, UK: Blackweel Scientific Publications.

Holzer, A., & Ondrus, J. (2011). Mobile Application Market: A Developer's Perspective. *Telematics and Informatics*, *28*, 22–31.

Hongladarom, S. (2002). The Web of Time and the Dilemma of Globalization. *The Information Society*, *18*, 241–249.

Hoven, J. van den. (2012). Human Capabilities and Technology. In I. Oosterlaken & J. van den Hoven (Eds.), *The Capability Approach, Technology and Design* (pp. 27–36). Springer Netherlands.

Huen, C. W. (2009). What is Context? An Ethnophilosophical Account. *Anthropological Theory*, *9*, 149–169.

Hutchby, I. (2001). Technologies, Texts and Affordances. *Sociology*, *35*, 441–456.

Iacob, C., Harrison, R., & Faily, S. (2013). Online Reviews as First Class Artifacts in Mobile App Development. In *Mobile Computing, Applications, and Services* (pp. 47–53). Springer, Cham.

Ibrahim-Dasuki, S., Abbott, P., & Kashefi, A. (2012). The Impact of ICT Investments on Development Using the Capability Approach: The case of the Nigerian Pre-paid Electricity Billing System. *The African Journal of Information Systems*, *4*. Retrieved from http://digitalcommons.kennesaw.edu/ajis/vol4/iss1/2

InfoDev. (2014). *Developing a Mobile Applications Sector in Afghanistan | infoDev*. Retrieved from http://www.infodev.org/articles/developing-mobile-applications-sector-afghanistan

International Telecommunications Union. (2017). *Uganda Profile*. Retrieved from https://www.itu.int/net4/itu-d/icteye/CountryProfileReport.aspx?countryID=240

Irani, L., Vertesi, J., Dourish, P., Philip, K., & Grinter, R. E. (2010). Postcolonial Computing: A Lens on Design and Development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1311–1320). New York, NY, USA: ACM.

James, J. (2006). Information Technology and Mass Poverty. *International Journal of Development Issues*, *5*, 85–107.

James, J. (2007). From Origins to Implications: Key Aspects in the Debate over the Digital Divide. *Journal of Information Technology*, *22*, 284–295.

James, J. (2010). New Technology in Developing Countries: A Critique of the One-Laptop-Per-Child Program. *Social Science Computer Review*, *28*, 381–390.

Jensen, R. (2007). The Digital Provide: Information (Technology), Market Performance, and Welfare in the South Indian Fisheries Sector. *The Quarterly Journal of Economics*, *122*, 879–924.

Jiménez, A., & Zheng, Y. (2017). Tech Hubs, Innovation and Development. *Information Technology for Development*, *0*, 1–24.

Johnstone, J. (2007). Technology as empowerment: a capability approach to computer ethics. *Ethics and Information Technology*, *9*, 73–87.

Joorabchi, M. E., Mesbah, A., & Kruchten, P. (2013). Real Challenges in Mobile App Development. In *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 15–24).

Jorgenson, D. W., Ho, M. S., & Stiroh, K. J. (2005). *Productivity, Volume 3: Information Technology and the American Growth Resurgence*. Massachusetts, USA: The MIT Press. Retrieved from https://ideas.repec.org/b/mtp/titles/0262101114.html

Juma, C., & Agwara, H. (2006). Africa in the Global Knowledge Economy: Strategic Options. *International Journal of Technology and Globalisation*, *2*, 220–233.

Kalan, J. (2014). Inside East Africa's Technology Hubs. Retrieved July 26, 2017, from http://www.bbc.com/future/story/20121018-inside-africas-technology-hubs

Kallinikos, J. (2010). *Governing Through Technology*. Hampshire, UK: Palgrave Macmillan Ltd.

Kallinikos, J., Aaltonen, A., & Marton, A. (2013). The Ambivalent Ontology of Digital Artifacts. *Management Information Systems Quarterly*, *37*, 357–370.

Kallinikos, J., Hasselbladh, H., & Marton, A. (2013). Governing Social Practice. *Theory and Society*, *42*, 395–421.

Kallinikos, J., & Mariátegui, J.-C. (2011). Video as Digital Object: Production and Distribution of Video Content in the Internet Media Ecosystem. *The Information Society*, *27*, 281–294.

Kambhampati, U. S. (2002). The Software Industry and Development: The Case of India. *Progress in Development Studies*, *2*, 23–45.

Karippacheril, T. G., Nikayin, F., De Reuver, M., & Bouwman, H. (2013). Serving the Poor: Multisided Mobile Service Platforms, Openness, Competition, Collaboration and the Struggle for Leadership. *Telecommun. Policy*, *37*, 24–34.

Kemp, R. (2002). Environmental Protection through Technological Regime Shifts. In A. Jamison & H. Rohracher (Eds.), *Technology Studies and Sustainable Development* (pp. 95–126). München: Profil Verlag.

Khan, F., & Ghadially, R. (2010). Empowerment through ICT Education, Access and Use: A Gender Analysis of Muslim Youth in India. *Journal of International Development*, *22*, 659–673.

Kimaro, H. C., & Nhampossa, J. L. (2005). Analyzing the problem of unsustainable health information systems in less-developed economies: Case studies from Tanzania and Mozambique. *Information Technology for Development*, *11*, 273–298.

King, G., Keohane, R. O., & Verba, S. (1994). *Designing Social Inquiry: Scientific Inference in Qualitative Research*. Princeton University Press.

Kitchin, R., & Dodge, M. (2011). *Code/Space*. Cambridge, UK: MIT Press. Retrieved from https://mitpress.mit.edu/books/codespace

Klecun, E., Lichtner, V., Cornford, T., & Petrakaki, D. (2014). Evaluation as a Multi-Ontological Endeavour: A Case from the English National Program for IT in Healthcare. *Journal of the Association for Information Systems*, *15*.

Klein, H. K. (2004). Seeking the New and the Critical in Critical Realism: Déjà Vu? *Information and Organization*, *14*, 123–144.

Kleine, D. (2010). ICT4WHAT?—Using the choice framework to operationalise the capability approach to development. *Journal of International Development*, *22*, 674–692.

Kshetri, N. (2016). Creation, Deployment, Diffusion and Export of Sub-Saharan Africa-Originated Information Technology-Related Innovations. *International Journal of Information Management*, *36*, 1274–1287.

Kvale, S., & Brinkmann, S. (2008). *InterViews: Learning the Craft of Qualitative Research Interviewing* (Second Edition edition). Los Angeles: SAGE Publications, Inc.

Lacity, M., Carmel, E., & Rottman, J. (2011). Rural Outsourcing: Delivering ITO and BPO Services from Remote Domestic Locations. *Computer*, *44*, 55–62.

Lange, C. (1921). Christian Lange - Nobel Lecture: Internationalism. Retrieved October 26, 2017, from https://www.nobelprize.org/nobel_prizes/peace/laureates/1921/lange-lecture.html

Lanvin, B., & Kralik, M. (2009). E-Skills: Who Made That Big Dent in My Flat World? *Information Technologies & International Development*, *5*, 81–84.

Latour, B. (1990). Technology is society made durable. *The Sociological Review*, *38*, 103–131.

Latour, B. (1992). Where are the missing masses? The sociology of a few mundane artefacts. In W.E Bijker & J. Law (Eds.), *Shaping technology building society: studies in sociotechnical change*. Cambridge, MA, US: The MIT Press. Retrieved from http://www.bruno-latour.fr/sites/default/files/50-MISSING-MASSES-GB.pdf

Latour, B. (2005). Reassembling the Social - An Introduction to Actor-Network-Theory. *Reassembling the Social - An Introduction to Actor-Network-Theory, by Bruno Latour, Pp. 316. Foreword by Bruno Latour. Oxford University Press, Sep 2005. ISBN-10: 0199256047. ISBN-13: 9780199256044*, *1*. Retrieved from http://adsabs.harvard.edu/abs/2005reso.book.....L

Lee, H., & Whitley, E. A. (2002). Time and Information Technology: Temporal Impacts on Individuals, Organizations, and Society. *The Information Society*, *18*, 235–240.

Leonardi, P. M. (2011). When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies. *MIS Quarterly*, *35*, 147–168.

Leonardi, P. M. (2013). Theoretical foundations for the study of sociomateriality. *Information and Organization*, *23*, 59–76.

Leonardi, P. M., & Barley, S. R. (2010). What's Under Construction Here? Social Action, Materiality, and Power in Constructivist Studies of Technology and Organizing. *The Academy of Management Annals*, *4*, 1–51.

Lessig, L. (2006). *Code: And Other Laws of Cyberspace, Version 2.0*. Cambridge, MA, US: Basic Books.

Lévi-Strauss, C. (1966). *The Savage Mind*. University of Chicago Press. Retrieved from http://www.press.uchicago.edu/ucp/books/book/chicago/S/bo28069706.html

Limayem, M., Hirt, S. G., & Cheung, C. M. K. (2007). How Habit Limits the Predictive Power of Intention: The Case of Information Systems Continuance. *MIS Quarterly*, *31*, 705–737.

Little, S. E. (2004). *Design and Determination: The Role of Information Technology in Redressing Regional Inequities in the Development Process*. Farnham, Surrey, United Kingdom: Ashgate.

Mackenzie, A. (2001). The Technicity of Time. *Time & Society*, *10*, 235–257.

Mackenzie, A. (2003). *Transduction: Invention, Innovation and Collective Life*. UK: Institute for Cultural Research, University of Lancaster. Retrieved from http://www.lancaster.ac.uk/staff/mackenza/papers/transduction.pdf

Mackenzie, A. (2005). Problematising the Technological: The Object as Event? *Social Epistemology*, *19*, 381–399.

Mackenzie, A. (2006). *Transductions: Bodies and Machines at Speed*. London, UK: A&C Black.

Macome, E. (2008). On implementation of an information system in the Mozambican context: The EDM case viewed through ANT lenses. *Information Technology for Development*, *14*, 154–170.

Madon, S. (1994). *Designing information systems for development planning*. Henley-on-Thames, UK: Alfred Waller Ltd. Retrieved from http://eprints.lse.ac.uk/27321/

Madon, S. (2004). Evaluating the Developmental Impact of E-Governance Initiatives: An Exploratory Framework. *The Electronic Journal of Information Systems in Developing Countries*, *20*, 1–13.

Madon, S., & Sharanappa, S. (2013). Social IT outsourcing and development: theorising the linkage. *Information Systems Journal*, *23*, 381–399.

Mandel, M. (2012). *Where the Jobs Are: The App Economy*. Retrieved from http://www.technet.org/wp-content/uploads/2012/02/TechNet-App-Economy-Jobs-Study.pdf

Mann, L., & Graham, M. (2016). The Domestic Turn: Business Process Outsourcing and the Growing Automation of Kenyan Organisations. *The Journal of Development Studies*, *52*, 530–548.

Margolin, V. (2007). Design for development: Towards a History. *Design Studies*, *28*, 111–115.

Margolin, V., & Margolin, S. (2002). A “Social Model” of Design: Issues of Practice and Research. *Design Issues*, *18*, 24–30.

Mariscal, J., Botelho, A. J. J., & Gutierrez, L. (2009). Training on Communication and Information Technologies, Employment and Youth: The Case of Brazil, Colombia, and Mexico. *Information Technologies & International Development*, *5*, 19–30.

Markides, C. (1997). Strategic Innovation. *MIT Sloan Management Review*, *38*.

Markman, G. D., Russo, M., Lumpkin, G. T., Jennings, P. D. (Dev), & Mair, J. (2016). Entrepreneurship as a Platform for Pursuing Multiple Goals: A Special Issue on Sustainability, Ethics, and Entrepreneurship. *Journal of Management Studies*, *53*, 673–694.

Markus, M. L., & Silver, M. (2008). A Foundation for the Study of IT Effects: A New Look at DeSanctis and Poole's Concepts of Structural Features and Spirit. *Journal of the Association for Information Systems*, *9*.

Marshall, C., & Rossman, G. B. (2006). *Designing Qualitative Research*. Sage Publications.

Masiero, S. (2015). Redesigning the Indian Food Security System through e-Governance: The Case of Kerala. *World Development*, *67*, 126–137.

Masiero, S. (2016). The Origins of Failure: Seeking the Causes of Design–Reality Gaps. *Information Technology for Development*. Retrieved from http://www.tandfonline.com/loi/titd20#.VwZ3FvkrLGg

Maslow, A. H. (1966). *The Psychology of Science*. New York: Harper & Row.

Mason, S. M., & Hacker, K. L. (2003). Applying Communication Theory to Digital Divide Research. *Information Technology and Society*, *5*, 40–55.

Matviyenko, S., & Miller, P. D. (2014). *The imaginary app*. Cambridge, Massachusetts, USA: The MIT Press.

Maurer, C., & Tiwana, A. (2012). Control in App Platforms: The Integration-Differentiation Paradox. *ICIS 2012 Proceedings*.

McGrath, K. (2013). The Potential of Generative Mechanisms for IS Research. *ICIS 2013 Proceedings*.

McGrath, K. (2016). Identity Verification and Societal Challenges: Explaining the Gap Between Service Provision and Development Outcomes. *MIS Quarterly*, *40*, 485–500.

Meagher, K., Mann, L., & Bolt, M. (2016). Introduction: Global Economic Inclusion and African Workers. *The Journal of Development Studies*, *52*, 471–482.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. London, UK: Sage Publishing.

Mingers, J. (2004). Real-izing information systems: critical realism as an underpinning philosophy for information systems. *Information and Organization*, *14*, 87–103.

Miscione, G. (2007). Telemedicine in the Upper Amazon: Interplay with Local Health Care Practices. *Management Information Systems Quarterly*, *31*.

Molla, A. (2000). Downloading or Uploading? The Information Economy and Africa's Current Status. *Information Technology for Development*, *9*, 205–221.

Morgan, K. (2004). The Exaggerated Death of Geography: Learning, Proximity and Territorial Innovation Systems. *Journal of Economic Geography*, *4*, 3–21.

Mubarak, F. (2015). Towards a Renewed Understanding of the Complex Nerves of the Digital Divide. *Journal of Social Inclusion*, *6*, 71–102.

Murenzi, R., & Hughes, M. (2006). Building a Prosperous Global Knowledge Economy in Africa: Rwanda as a Case Study. *International Journal of Technology and Globalisation*, *2*, 252–267.

Mutch, A. (2009). Technology, Organization, and Structure—A Morphogenetic Approach. *Organization Science*, *21*, 507–520.

Mutch, A. (2013). Sociomateriality — Taking the Wrong Turning? *Information and Organization*, *23*, 28–40.

Myers, M. D. (2004). Social Theory and Philosophy for Information Systems. In John Mingers & L. Willcocks (Eds.), *Social theory and philosophy for information systems*. J. Wiley.

Myers, M. D., & Newman, M. (2007). The Qualitative Interview in IS Research: Examining the Craft. *Information and Organization*, *17*, 2–26.

Napoli, P. M., & Karppinen, K. (2013). Translating Diversity to Internet Governance. *First Monday*, *18*.

Napoli, P. M., & Obar, J. A. (2014). The Emerging Mobile Internet Underclass: A Critique of Mobile Internet Access. *The Information Society*, *30*, 323–334.

Negroponte, N. (1995). *Being digital*. Vintage Books.

Nelson, R. R. (1982). *An Evolutionary Theory of Economic Change*. Belknap Press of Harvard University Press.

Nieusma, D. (2004). Alternative Design Scholarship: Working Toward Appropriate Design. *Design Issues*, *20*, 13–24.

Nieusma, D., & Riley, D. (2010). Designs on development: engineering, globalization, and social justice. *Engineering Studies*, *2*, 29–59.

Njihia, J. M., & Merali, Y. (2013). The Broader Context for Ict4d Projects: A Morphogenetic Analysis. *MIS Quarterly*, *37*, 881–905.

Norman, D. A. (1999). Affordance, Conventions, and Design. *Interactions*, *6*, 38–43.

Ojanperä, S., Graham, M., Straumann, R., Sabbata, S. D., & Zook, M. (2017). Engagement in the Knowledge Economy: Regional Patterns of Content Creation with a Focus on Sub-Saharan Africa. *Information Technologies & International Development*, *13*, 19.

Oosterlaken, I. (2009). Design for Development: A Capability Approach. *Design Issues*, *25*, 91–102.

Oosterlaken, I. (2012). The Capability Approach, Technology and Design: Taking Stock and Looking Ahead. In I. Oosterlaken & J. van den Hoven (Eds.), *The Capability Approach, Technology and Design* (pp. 3–26). Springer Netherlands.

Orlikowski, W. J. (2000). Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. *Organization Science*, *11*, 404–428.

Orlikowski, W. J. (2007). Sociomaterial Practices: Exploring Technology at Work. *Organization Studies*, *28*, 1435–1448.

Orlikowski, W. J., & Iacono, C. S. (2001). Research Commentary: Desperately Seeking the "IT" in IT Research—A Call to Theorizing the IT Artifact. *Information Systems Research*, *12*, 121–134.

Orlikowski, W. J., & Scott, S. V. (2008). Sociomateriality: Challenging the Separation of Technology, Work and Organization. *The Academy of Management Annals*, *2*, 433–474.

Oxford Dictionary. (2015). Context. Retrieved October 11, 2015, from http://www.oxforddictionaries.com/definition/english/context

Ozcan, P., & Santos, F. M. (2015). The Market That Never Was: Turf Wars and Failed Alliances in Mobile Payments. *Strategic Management Journal*, *36*, 1486–1512.

Papanek, V. (1983). For the Southern Half of the Globe. *Design Studies*, *4*, 61–64.

Papanek, V. (1986). Design in Developing Countries 1950 – 1985: A Summing-Up. *Art Libraries Journal*, *11*, 44–51.

Parayil, G. (2005). The Digital Divide and Increasing Returns: Contradictions of Informational Capitalism. *Information Society*, *21*, 41–51.

Parmigiani, A., & Howard-Grenville, J. (2011). Routines Revisited: Exploring the Capabilities and Practice Perspectives. *Academy of Management Annals*, *5*, 413–453.

Patibandla, M., & Petersen, B. (2002). Role of Transnational Corporations in the Evolution of a High-Tech Industry: The Case of India's Software Industry. *World Development*, *30*, 1561–1577.

Patton, A. (2016, February 16). Uganda Is a Land of Entrepreneurs, but How Many Startups Survive? *The Guardian*. Retrieved from http://www.theguardian.com/global-development-professionals-network/2016/feb/16/uganda-is-a-land-of-entrepreneurs-but-how-many-startups-survive

Pedersen, P. E., & Ling, R. (2002). Mobile End-User Service Adoption Studies: A Selective Review. *Scandinavian Journal of Information Systems*, *14*, 3–17.

Pentland, B. T. (1995). Grammatical Models of Organizational Processes. *Organization Science*, *6*, 541–556.

Pentland, B. T., & Rueter, H. H. (1994). Organizational Routines as Grammars of Action. *Administrative Science Quarterly*, *39*, 484–510.

Pettigrew, A. M. (1985). Contextualist Research and the Study of Organizational Change Processes. In E. Mumford, R. Hirschheim, G. Fitzgerald, & A. T. Wood-Harper (Eds.),

Research Methods in Information Systems (pp. 53–78). Amsterdam, The Netherlands: North-Holland Publishing Co.

Pew Research Center. (2015). *Cell Phones in Africa: Communication Lifeline Texting Most Common Activity, but Mobile Money Popular in Several Countries*. Washington, DC: Pew Research Center.

Pimienta, D. (2008). Accessing Content. In F. Alan & L. Nordstrom (Eds.), *Global Information Society Watch*. Melville, South Africa: Alliance for Progressive Communications, Hivos, and ITeM. Retrieved from http://www.giswatch.org/thematic-report/2008-access-infrastructure/accessing-content

Pinch, T. J., & Bijker, W. E. (1984). The Social Construction of Facts and Artefacts: or How the Sociology of Science and the Sociology of Technology might Benefit Each Other. *Social Studies of Science*, *14*, 399–441.

Powell, M. (2001). Knowledge, Culture & the Internet in Africa: A Challenge for Political Economists. *Review of African Political Economy*, *28*, 241–260.

Prakash, A., & De', R. (2007). Importance of development context in ICT4D projects: A study of computerization of land records in India. *Information Technology & People*, *20*, 262–281.

Puri, S. K., & Sahay, S. (2003). Institutional Structures and Participation. In M. Korpela, R. Montealegre, & A. Poulymenakou (Eds.), *Organizational Information Systems in the Context of Globalization* (pp. 271–287). Springer US.

Rabinow, P. (2002). *French DNA: Trouble in Purgatory*. Chicago, USA: University of Chicago Press.

Rahim, S. A., Pawanteh, L., & Salman, A. (2011). Digital Inclusion: The Way Forward for Equality in a Multiethnic Society. *Innovation Journal*, *16*.

Rahimian, V., & Habibi, J. (2008). Performance Evaluation of Mobile Software Systems: Challenges for a Software Engineer. In *5th International Conference on Electrical Engineering, Computing Science and Automatic Control, 2008. CCE 2008* (pp. 346–351).

Rai, S., Harindranath, G., & Liebenau, J. (2013). Special Issue on IT Innovation in Emerging Economies. *Journal of Information Technology*, *28*, 261–263.

Rice, R. E., & Katz, J. E. (2003). Comparing Internet and Mobile Phone Usage: Digital Divides of Usage, Adoption, and Dropouts. *Telecommunications Policy*, *27*, 597–623.

Ritzer, G. (2007). *The McDonaldization of Society 5* (5th edition). Los Angles, Calif: SAGE Publications, Inc.

Roberts, F. M. (1989). The Finnish Coffee Ceremony and Notions of Self. *Arctic Anthropology*, *26*, 20–33.

Robertson, P. R. (1992). *Globalization: Social Theory and Global Culture*. SAGE.

Robey, D., Boudreau, M.-C., & Rose, G. M. (2000). Information Technology and Organizational Learning: a Review and Assessment of Research. *Accounting, Management and Information Technologies*, *10*, 125–155.

Robeyns, I. (2005). The Capability Approach: a Theoretical Survey. *Journal of Human Development*, *6*, 93–117.

Robson, C. (2011). *Real World Research* (3rd Edition). John Wiley & Sons, UK.

Rogers, E. M. (2003). *Diffusion of Innovations, 5th Edition* (5 edition). New York: Free Press.

Rolland, K. H., & Monteiro, E. (2002). Balancing the Local and the Global in Infrastructural Information Systems. *The Information Society*, *18*, 87–100.

Rothschild, J. (Ed.). (1999). *Design and Feminism: Revisioning Spaces, Places and Everyday Things*. New Brunswick, N.J: Rutgers University Press.

Rubin, H. J., & Rubin, I. S. (2011). *Qualitative Interviewing: The Art of Hearing Data*. Thousand Oaks, California,USA: SAGE.

Ryan, C., & Gonsalves, A. (2005). The Effect of Context and Application Type on Mobile Usability: An Empirical Study. In *Proceedings of the Twenty-eighth Australasian Conference on Computer Science - Volume 38* (pp. 115–124). Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

Ryn, S. V. der, & Cowan, S. (1996). *Ecological Design*. London, UK: Island Press.

Sandeep, M. S., & Ravishankar, M. N. (2016). Impact sourcing ventures and local communities: a frame alignment perspective. *Information Systems Journal*, *26*, 127–155.

Sayer, R. A. (2000). *Realism and Social Science*. London, UK: Sage.

Schejter, A. M., & Tirosh, N. (2012). Social Media New and Old in the Al-'Arakeeb Conflict: A Case Study. *The Information Society*, *28*, 304–315.

Schienstock, G. (2011). Path Dependency and Path Creation: Continuity vs. Fundamental Change in National Economies. *Journal of Futures Studies*, *15*, 63–76.

Schradie, J. (2011). The Digital Production Gap: The Digital Divide and Web 2.0 Collide. *Poetics*, *39*, 145–168.

Schuler, D., & Namioka, A. (Eds.). (1993). *Participatory Design: Principles and Practices*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Schware, R. (2009). Give For-Profit Rural Business Centers a Chance to Diversify Into Service-Led Employment and Village BPOs. *Information Technologies & International Development*, *5*, 77–80.

Scott, S. V., & Orlikowski, W. J. (2012). Great Expectations: The Materiality of Commensurability in Social Media. In P. M. Leonardi, B. A. Nardi, & J. Kallinikos (Eds.), *Materiality and Organizing* (pp. 113–133). Oxford, UK: Oxford University Press.

Searle, J. R. (1995). *The Construction of Social Reality*. London, UK: Penguin.

Sen, A. (1999). *Development as Freedom*. Oxford, UK: Oxford University Press.

Sen, A. (2010). The Mobile and the World. *Information Technologies & International Development*, *6*, 1–3.

Singer, H. W. (1970). Dualism Revisited: A New Approach to the Problems of the Dual Society in Developing Countries. *The Journal of Development Studies*, *7*, 60–75.

Sismondo, S. (1993). Some Social Constructions. *Social Studies of Science*, *23*, 515–553.

Smith, M. L. (2006). Overcoming Theory-Practice Inconsistencies: Critical Realism and Information Systems Research. *Information and Organization*, *16*, 191–211.

Smith, M. L., & Seward, C. (2009). The Relational Ontology of Amartya Sen's Capability Approach: Incorporating Social and Individual Causes. *Journal of Human Development and Capabilities*, *10*, 213–235.

Smith, M. L., Spence, R., & Rashid, A. T. (2011). Mobile Phones and Expanding Human Capabilities. *Information Technologies & International Development*, *7*, 77–88.

Sonenshein, S. (2016). Routines and Creativity: From Dualism to Duality. *Organization Science*, *27*, 739–758.

Souter, D. (2011). Mobile internet usage and demand in Kenya: The experience of early adopters. In *Making Broadband Accessible for All*. Newbury, UK: Vodafone. Retrieved from http://www.vodafone.com/content/dam/vodafone/about/public_policy/policy_papers/public_policy_series_12.pdf

Star, S. L., & Ruhleder, K. (1996). Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research*, *7*, 111–134.

Streeten, P. (1981). *Development Perspectives*. London, UK: Palgrave Macmillan. doi:10.1007/978-1-349-05341-4_19

Swanson, E., Minges, M., Clarke, G. R., Safdar, Z., Zhen-Wei Qiang, C., Adamali, A., … Halewood, N. (2006). *Information and Communications for Development* (No. 35924) (pp. 1–23). The World Bank. Retrieved from http://documents.worldbank.org/curated/en/692321468170348192/Overview

Tapia, A., & Maldonado, E. (2009). An ICT Skills Cascade: Government-Mandated Open Source Policy as a Potential Driver for ICT Skills Transfer. *Information Technologies & International Development*, *5*, 31–51.

Thomas, A. (2006). Design, Poverty, and Sustainable Development. *Design Issues*, *22*, 54–65.

Thompson, M. (2002). Cultivating Meaning: Interpretive Fine-Tuning of a South African Health Information System. *Information and Organization*, *12*, 183–211.

Thrane, S., Blaabjerg, S., & Møller, R. H. (2010). Innovative Path Dependence: Making Sense of Product and Service Innovation in Path Dependent Innovation Processes. *Research Policy*, *39*, 932–944.

Tilson, D., Lyytinen, K., & Sørensen, C. (2010). Research Commentary—Digital Infrastructures: The Missing IS Research Agenda. *Information Systems Research*, *21*, 748–759.

Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Research Commentary—Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. *Information Systems Research*, *21*, 675–687.

Toyama, K. (2011). Technology As Amplifier in International Development. In *Proceedings of the 2011 iConference* (pp. 75–82). New York, NY, USA: ACM.

Tsang, E. W. K. (2014). Case studies and generalization in information systems research: A critical realist perspective. *The Journal of Strategic Information Systems*, *23*, 174–186.

Tuomi, I. (2002). *Networks of Innovation: Change and Meaning in the Age of the Internet*. Oxford, New York: Oxford University Press.

Uganda Bureau of Statistics. (2016). *The National Population and Housing Census 2014 - Main Report*. Kampala, Uganda. Retrieved from https://unstats.un.org/unsd/demographic/sources/census/wphc/Uganda/UGA-2016-05-23.pdf

Uganda Ministry of ICT. (2015). ICT Sector Priorities | Ministry of ICT. Retrieved November 28, 2016, from http://www.ict.go.ug/ict/ict-sector-priorities

UNDP. (2013). Income Gini coefficient | Human Development Reports. Retrieved October 21, 2017, from http://hdr.undp.org/en/content/income-gini-coefficient

UNDP. (2016a). *Africa Human Development Report 2016*. Retrieved from http://www.undp.org/content/undp/en/home/librarypage/hdr/2016-africa-human-development-report.html

UNDP. (2016b). *Human Development Report 2016 - Country Notes Uganda*. Retrieved from http://hdr.undp.org/sites/all/themes/hdr_theme/country-notes/UGA.pdf

UNDP. (2017a). Human Development Reports - HDI. Retrieved July 27, 2017, from http://hdr.undp.org/en/composite/HDI

UNDP. (2017b). Human Development Reports - Uganda. Retrieved July 27, 2017, from http://hdr.undp.org/en/countries/profiles/UGA

US Embassy in Uganda Hackathon. (2015, December 29). Are you an entrepreneur? Do you want win $5,000? Retrieved from https://ug.usembassy.gov/are-you-an-entrepreneur-do-you-want-win-5000/

van Dijk, J. (2006). Digital Divide Research, Achievements and Shortcomings. *Poetics*, *34*, 221–235.

Verplanken, B., Aarts, H., & Van Knippenberg, A. (1997). Habit, Information Acquisition, and the Process of Making Travel Mode Choices. *European Journal of Social Psychology*, *27*, 539–560.

Vision 2030. (2007). Kenya Vision 2030 website. Retrieved November 28, 2016, from http://www.vision2030.go.ke/home/

von Hippel, E., & Katz, R. (2002). Shifting Innovation to Users via Toolkits. *Management Science*, *48*, 821–833.

Wagner, S., & Fernández-Ardèvol, M. (2016). Local Content Production and the Political Economy of the Mobile App Industries in Argentina and Bolivia. *New Media & Society*, *18*, 1768–1786.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an Information System Design Theory for Vigilant EIS. *Information Systems Research*, *3*, 36–59.

Walsham, G. (1995a). Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information Systems*, *4*, 74–81.

Walsham, G. (1995b). The Emergence of Interpretivism in IS Research. *Information Systems Research*, *6*, 376–394.

Walsham, G. (2001). *Making a World of Difference: IT in a Global Context*. Chichester, UK: Wiley.

Walsham, G. (2002). Cross-Cultural Software Production and Use: A Structurational Analysis. *MIS Quarterly*, *26*, 359–380.

Walsham, G. (2017). ICT4D Research: Reflections on History and Future Agenda. *Information Technology for Development*, *23*, 18–41.

Walsham, G., Robey, D., & Sahay, S. (2007). Foreword: Special Issue on Information Systems in Developing Countries. *MIS Quarterly*, *31*, 317–326.

Walsham, G., & Sahay, S. (2006). Research on Information Systems in Developing Countries: Current Landscape and Future Prospects. *Information Technology for Development*, *12*, 7–24.

Walton, R., Putnam, C., Johnson, E., & Kolko, B. (2009). Skills Are Not Binary: Nuances in the Relationship Between ICT Skills and Employability. *Information Technologies & International Development*, *5*, 1–18.

Wang, Y., Lai, P., & Sui, D. (2003). Mapping the Internet using GIS: The death of distance hypothesis revisited. *Journal of Geographical Systems*, *5*, 381–405.

Wareham, J., Levy, A., & Shi, W. (2004). Wireless Diffusion and Mobile Computing: Implications for the Digital Divide. *Telecommunications Policy*, *28*, 439–457.

Warschauer, M. (2002). Reconceptualizing the Digital Divide. *First Monday*, *7*.

Warschauer, M. (2003). Demystifying the Digital Divide. *Scientific American*, *289*, 42–47.

Waverman, L., Meschi, M., & Fuss, M. (2005). *The Impact of Telecoms on Economic Growth in Developing Countries* (The Vodafone Policy Paper Series No. 3) (pp. 10–23).

Wikipedia. (2014, November 1). Ushahidi. In *Ushahidi*. Retrieved from http://en.wikipedia.org/w/index.php?title=Ushahidi&oldid=622200358

Wilensky, H. L. (1967). *Organizational Intelligence*. New York: Basic Books.

Williams, R., & Edge, D. (1996). The Social Shaping of Technology. *Research Policy*, *25*, 865–899.

Winner, L. (1980). Do Artifacts Have Politics? *Daedalus*, *109*, 121–136.

Wohlstetter, R. (1962). *Pearl Harbor*. Stanford, CA, USA: Stanford University Press.

World Health Organization. (2017). Uganda:Issues and challenges - Progress on the Health-Related MDGs - AHO. Retrieved July 27, 2017, from http://www.aho.afro.who.int/profiles_information/index.php/Uganda:Issues_and_challenges_-_Progress_on_the_Health-Related_MDGs

Wu, T. (2007). Wireless Carterfone. *International Journal of Communication*, *1*, 389–426.

Wynn, D., & Williams, C. K. (2012). Principles for Conducting Critical Realist Case Study Research in Information Systems. *MIS Quarterly*, *36*, 787–810.

Xiao, X., Califf, C. B., Sarker, S., & Sarker, S. (2013). ICT Innovation in Emerging Economies: A Review of the Existing Literature and a Framework for Future Research. *Journal of Information Technology*, *28*, 264–278.

Ya'u, Y. Z. (2005). Globalisation, ICTs, and the New Imperialism: Perspectives on Africa in the Global Electronic Village (GEV). *Africa Development*, *30*.

Yin, R. K. (2014). *Case Study Research: Design and Methods (5th edition)*. Washington DC, USA: SAGE.

Yoo, Y., Boland, R. J. J., Lyytinen, K., & Majchrzak, A. (2012). Organizing for Innovation in the Digitized World. *Organization Science*, *23*, 1398–1408.

Yoo, Y., Henfridsson, O., & Lyytinen, K. (2010). Research Commentary—The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research. *Information Systems Research*, *21*, 724–735.

Zainudeen, A., & Ratnadiwakara, D. (2011). Are the Poor Stuck in Voice? Conditions for Adoption of More-Than-Voice Mobile Services. *Information Technologies & International Development*, *7*, 45–59.

Zammuto, R. F., Griffith, T. L., Majchrzak, A., Dougherty, D. J., & Faraj, S. (2007). Information Technology and the Changing Fabric of Organization. *Organization Science*, *18*, 749–762.

Zheng, Y. (2009). Different Spaces for e-Development: What Can We Learn from the Capability Approach? *Information Technology for Development*, *15*, 66–82.

Zhu, K., Kraemer, K. L., Gurbaxani, V., & Xu, S. X. (2006). Migration to Open-Standard Interorganizational Systems: Network Effects, Switching Costs, and Path Dependency. *MIS Quarterly*, *30*, 515–539.

Zittrain, J. L. (2008). *The Future of the Internet and How to Stop It*. New Haven, United States: Yale University Press.