Improving ICT4D projects with Agile software development

Student: Joost Dijkers (5527554)

Bachelor thesis
Bachelor: Information Science
University: Utrecht University

1st supervisor: dr. S.J. Overbeek
2nd supervisor: dr. S. España

Date: 27-6-2017

Abstract: ICT4D seeks to bridge the digital divide in developing countries. Often voiced requirements are that an ICT4D project needs to be demand-driven and value participation of the local community. User collaboration is a principle of Agile software development (Agile), thus suggesting that Agile can improve ICT4D projects. This thesis aims to answer to what degree Agile can improve ICT4D projects. In order to achieve this, existing literature was consulted and an interview was held. This thesis provides an overview of the critical success factors for ICT4D projects, as well as of the critical success factors for, advantages of, and limitations of Agile. Agile can only work successfully when the ICT4D project is demand-driven, and when both a cultural understanding and trust are built. The most important ways in which Agile can improve ICT4D projects are by facilitating user collaboration, improving team communication, enhancing organizational learning, and by frequently delivering software.

Keywords: Agile, CSF, ICT4D

Table of content

Acknowledgements

The thesis would not have been possible without the help of several people. The author wishes to thank dr. Overbeek for supervising the research project. Due to author's inexperience with writing scientific articles, there were many things the author was uncertain about. Dr. Overbeek would regularly provide feedback and put the author back on course whenever the author strayed off course. Additionally, the author wishes to thank dr. España for providing a second opinion and for being the second reviewer. Finally, the author wishes to thank prof. dr. Akkermans, ms. drs. Bon, and ms. drs. Tuijp for sharing their expertise during the interview.

List of tables

1. Introduction

The use of ICT in developing countries is the focus of an academic field called information and communication technology for development, or ICT4D in short (Walsham, 2017). Walsham states that ICT4D is aimed at how the benefits of ICT can be evenly divided between society to bridge the gap between the rich and poor. But high rates of failure exist for ICT4D projects (Independent Evaluation Group, 2011, p. 81). Agile Software Development (from now on simply Agile), which is a way of developing software, was found to increase the success rate of ICT projects (Serrador & Pinto, 2015, p. 1047). However, ICT4D projects have many exogenous and endogenous variables that need to be closely considered. For example, political powers can take control of and obstruct the project (Ferguson and Ballantyne, 2002). Another example is that the people in poor communities often have no ICT or project management skills (James, 2004). It is possible that a characteristic of Agile can improve taking such variables into account. However, it is also possible that such variables can make using Agile methods more difficult, or less beneficial. Ciaghi & Villafiorita (2011) also warn against carelessly using Agile methods that have been successful in western countries (p. 171). Research is thus necessary to determine the full effects of using the Agile approach in an ICT4D project. The aim of this thesis is to find an answer to the following question: "To what degree can Agile software development improve ICT4D projects?". But first, more background information on ICT4D and Agile will be given, as well as more reasons why this research question is important.

ICT can play three important roles in development (Hanna, 2010). First, ICT allows for the access to and processing of information and knowledge at an unprecedented speed. As a result, the creation, sharing, and improving of knowledge can enhance. Secondly, ICT speeds up production and transactions as well as reduces the costs. Finally, ICT is an effective tool to create and build connections among different parties, such as firms or communities.

ICT4D underwent several important changes (Heeks, 2008a). The first phase lasted from 1956 until the 1990s, and is referred to as ICT4D 0.0. During ICT4D 0.0 the focus was on how ICT can be used by the government for mainly administrative purposes, and later by the private sector to spur economic growth. In the 1990s two important advancements occurred that changed ICT4D. First, the rise of the Internet gave birth to new opportunities in which ICT can be used to benefit developing countries. Secondly, the United Nations adopted the Millennium Declaration in which goals such as reducing poverty and improving health were defined (p. 27). And so the transition to ICT4D 1.0 was set in motion, in which the focus lied on how international development organizations and nongovernmental organizations sought to use ICT to address the goals from the Millennium Declaration. Several issues arose: sustainability, scalability, and objective evaluation of projects. Providing solutions to those areas marks the transition from ICT4D 1.0 to 2.0. Several practical solutions include providing local content (p. 28), creating employment (p. 29), and working alongside the poor communities (p. 30). The most important difference between ICT4D 1.0 and 2.0 however, is that 1.0 was supply-driven, whereas 2.0 is demand-driven and actively seeks participation from the poor in ICT4D projects. Bon and Akkermans (2014) propose an evolution from ICT4D 2.0 to ICT4D 3.0 (p. 2). ICT4D 3.0 is about complex, adaptive systems, which Monge & Contractor (2003) describe as systems in which a large number of participants with different attributes move independently according to their own rules and interact with each other. According to Bon and Akkermans, ICT4D involves interaction between different participants (for example, the beneficiaries), with requirements and knowledge as a result of that interaction. Participation in the development process of those participants is very important.

An important element of ICT4D is the digital divide, which De Munster (2005) defines as: "A gap […] between those individuals that can access new information and communication tools […] and those

who are too poor to get them" (p. 133). De Munster stresses that it is important to not just provide access to those who cannot obtain the information and communication tools, but also to train the people in using them.

An important difference between 'normal' and ICT4D projects must be raised. An interview was held with ICT4D experts (see chapter 2.5, Appendix B), and according to the interviewees the customer and the user are different parties in an ICT4D project, unlike in 'normal' ICT projects. In an ICT4D project, the donor is the customer, while the local community is the user. So whereas customer and user can normally be used interchangeably, in this thesis these two terms will from here on out mean different things.

Taking a step back, ICT is developed using a specific software development approach. Agile is such an approach and it is a philosophy on which some methods are based. In 2001 a group of software development experts came together and created the foundation of Agile: The Agile Manifesto. On their website the values behind Agile are explained as follows ("Manifesto for Agile software development", 2001):

> "Individuals and interactions over processes and tools
> Working software over comprehensive documentation
> Customer collaboration over contract negotiation
> Responding to change over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more."

Paulk (2002) argues that the values from The Agile Manifesto create synergy together. Paulk further argues that any Agile method that lacks one or more of these values is doomed to fail.

Agile is collaborative, incremental, iterative, and adaptive. Collaborative development (Ambler, 2005, p. 35) means that work is performed in teams rather than individually. For Agile, this also means that users should be included in the work process. Incremental development is a method in which the system is developed in a series of small steps (Ambler, p. 35). Iterative development means that the development activities, such as requirement engineering and software testing, are performed cyclically rather than sequentially (Ambler, p. 35). Finally, adaptive development is a method in which (rapid) change is supported (Highsmith, 2002, p. 7). Highsmith summarises the practices of Agile as follows: "*short iterations, continuous testing, selforganizing teams, constant collaboration (…), and frequent replanning based on current reality*" (p. 4).

An example of an Agile method is Scrum. In Scrum (Rising & Janoff, 2000, p. 30), the project is first broadly planned while still allowing for change throughout the development. Then, the product is developed incrementally in short phases called sprints. During these sprints frequent meetings are held (normally daily). Each sprint builds the system incrementally. Customer feedback is allowed in between sprints.

ICT4D 2.0 is demand-driven with a focus on participation of the poor community. Agile, then, would be a good fit due to user collaboration being a central pillar. Furthermore, this focus on user collaboration was found to increase user satisfaction (Ceschi, Sillitti, Succi, & De Panfilis, 2005, p. 24), which, in turn, was found to increase the success rate of software development projects (Serrador & Pinto, 2015, p. 1047). Additionally, Agile may play an important role if the transition from ICT4D 2.0 to 3.0 becomes widely accepted: because of the iterative and collaborative nature of Agile methods, Bon and Akkermans (2014) believe that purposeful and adaptive artefacts can be created (p. 2).

The structure of this thesis is as follows. First, in chapter 2 the details of the research approach will be given. Second, in chapter 3, related work about the use of Agile in ICT4D projects will be covered. The goal of this is to determine what is known about this subject, and what the value of this thesis is. Subsequently, in chapter 4, chapter 5, and chapter 6 the sub questions as defined in chapter 2.1 will be answered. Then, in chapter 7 the benefits and limitations of this thesis will be discussed. Finally, in chapter 8 the most important findings will be summarised and an answer will be given to the research question. Additionally, suggestions for further research will be given. Appendix A describes the execution of the literature review as specified in chapter 2.4, and Appendix B describes the structure of the interview as specified in chapter 2.5.

2. Research approach

In this chapter the approach that is used to find an answer to the main research questions will be covered. First, the research question will be divided into sub questions. Then, the method for gathering data will be explained. Afterwards, the criteria for selecting data will be specified, and the way the selected data was used to answer the sub questions. Fourth, the method for finding related works in Agile methods for ICT4D will be described. Finally, the details of the interview will be specified.

2.1. Main research question and sub questions

This thesis aims to answer the main research question: "To what degree can Agile software development improve ICT4D projects?". It will do so by answering the following sub questions (SQ):

SQ1: *What are the critical success factors for ICT4D projects?*

In order to understand the importance of SQ1, it is important to introduce the concept of critical success factors. Alias, Zawawi, Yusof, and Aris (2014) define critical success factors as: "Inputs to project management practice which can lead directly or indirectly to project success" (p. 61). As such, critical success factors give a good impression of where Agile might have an important effect. SQ1 will be covered in chapter 4.

SQ2: *What are the critical success factors, advantages, and limitations for Agile software development?*

By looking at the advantages and limitations of Agile, an answer can be given to what aspects of Agile might have an important effect on ICT4D. Additionally, SQ2 will give insight into what is necessary to make Agile work successfully, which is required for software development projects to succeed (Chow & Cao, 2007, p. 962). SQ2 will be covered in chapter 5.

SQ3: *Can Agile methods successfully work in an ICT4D project?*

An analysis will be performed to determine if Agile can work successfully in ICT4D projects. The purpose of this sub question follows the same argumentation as to why the critical success factors for Agile are summarized for SQ2: it is essential for Agile to work successfully in order for the ICT4D project to succeed (Chow &Cao, 2007, p. 962). SQ3 will be covered in chapter 6.1.

SQ4: *How can ICT4D projects benefit from an Agile approach?*

The benefits Agile can bring specifically to ICT4D projects need to be discussed in order to understand how Agile can improve ICT4D projects. SQ4 will be covered in chapter 6.2.

2.2. Data gathering

These sub questions are answered by using existing literature. To search for literature the search engine Google Scholar was primarily used. IEEE Computer Society Digital Library and Springer Link were also used, but to a lesser degree because using Google Scholar often sufficed.

Some common search terms that were used are: "Agile software development", "Agile software development benefits", "Agile software development critical success factors", "ICT4D", "ICT4D

critical success factor", "ICT4D case", and "ICT4D Agile". One notable search term that was used less frequently is "ICTD", which is an alternative way of referring to ICT4D. Sometimes more information was required on a specific subject, resulting in an one time search term. An example is "Distributed development", which was used to give a definition of distributed development in chapter 5.3. Another way of searching for literature that was used is the forward snowball method, which is searching for new papers in the reference list of papers already found (Glock, 2017).

2.3. Data analysis

In order to answer the first sub question the gathered data was selected based on whether or not it could be considered a critical success factor. The following terms were found and used (for ICT4D projects): critical success factor, lesson learned, step (to ensure sustainable development), and activity (that led to success). Additionally, for the first sub question three case studies performed by other researchers were summarized and analysed. The objective of this was to see how the found critical success factors play a role in a real ICT4D project, as well as to find potential new critical success factors. The three case studies discuss the following projects: The Happy Valley Project (Krauss, 2016), the Sichuan rural informatization program (Liu, 2016), and Impact Venture (Sandeep & Ravishankar, 2016). These three case studies were selected on two criteria. First, the case studies had to be recent (2016 or 2017) to ensure that the knowledge of ICT4D applied was as up-to-date as possible. Second, the case studies had to be different in nature to ensure variety in the results. The Happy Valley Project is about ICT training, the Sichuan rural informatization program is about the sustainability of said program, and Impact Venture is about a business process outsourcing vendor firm seeking to create jobs in disadvantaged communities.

In order to answer the second sub question three categories of information were necessary: critical success factors, advantages, and limitations. For Agile, the following two terms were found and used to describe critical success factors: critical success factor, lesson learned. Advantages were sometimes also called benefits. Limitations were sometimes called disadvantages.

The third and fourth sub questions were answered by using the results from the first two sub questions. More specifically, first a comparison between the critical success factors for ICT4D projects and for Agile was performed. This resulted in an overview of which critical success factors for ICT4D projects would need to be satisfied before Agile can work. Then, in order to determine how Agile can improve ICT4D projects, a comparison between the advantages of Agile and the critical success factors for ICT4D projects was made.

2.4. Method for finding related work in Agile methods for ICT4D

To search for related works in Agile methods for ICT4D, a literature review was performed. A literature review seeks to provide a comprehensive overview of the knowledge on a specific topic (Green, Johnson, & Adams, 2006, p. 101). Multiple types of literature reviews exist. One of them is the systematic literature review, which involves rigorously using a step-by-step method to search for literature (p. 104).

One framework for systematic literature review is the Preferred Reporting Items for Systematic Reviews and Meta-Analyses, or the PRISMA Statement for short (Moher, Liberati, Tetzlaff, Altman, & Prisma Group, 2009). The PRISMA Statement suggests four phases for systematic literature review. First is the identification phase, in which articles are searched by using search engines. Then, there is the screening phase, in which duplicates are removed and the articles are screened. Third, an eligibility phase is conducted, in which the eligibility of the remaining articles is determined. Finally,

there is an included phase, which results in a comprehensive list of the included articles. Siddaway (n.d.) suggests that during the screening phase articles are judged based on the title and abstract (p. 7), and that during the eligibility phase articles are judged based on the full text (p. 7). The PRISMA Statement also suggests a checklist for structuring systematic literature reviews (Moher et al., 2009). For example, the title should mention that the paper is a systematic literature review. Another example is that the risk of bias among the included studies should be analysed.

The literature review performed in this thesis differs from the PRISMA Statement in several ways due to time constraints. First, the literature review was not exhaustive. As such, only the first ten results from each query were used. Second, not all the checklist items from the PRISMA Statement were considered. Because the literature review performed in this thesis is different from the PRISMA Statement, it cannot be classified as a systematic literature review.

In Table 1 the PRISMA Statement checklist items that were considered are mentioned and explained. The table includes in the following order: the objective of the literature review, the criteria used to determine the eligibility of the articles, the search engines used, the way articles were sought, the selection process of studies, the processing of collecting information from the included articles, and, finally, a summary of the relevant results from the included articles. The numbers correspond with the numbers on the checklist.

PRISMA Statement checklist item	Explanation
4. Objective	The objective of the literature review is to gain insight on existing research related to Agile methods for ICT4D. In particular to what degree the question of how Agile can improve ICT4D projects has already been researched.
6. Eligibility criteria	Eligible articles are categorized in two different ways: articles that talked about benefits from Agile as experienced in ICT4D projects and articles that propose an Agile framework or a framework with Agile inspirations specifically developed for ICT4D projects.
7. Information sources	Three search engines were used: Google Scholar, IEEE Computer Society Digital Library, and Springer Link.
8. Search	Multiple queries were designed, which were used for all search engines. Each query consists of three parts: Agile, ICT4D, and method. For Agile, both “Agile” and “Agile software development” were used. For the ICT4D, “ICT4D” was used as a term. “ICTD”, which is another way of referring to ICT4D, was also used. Finally, for the method part both “method” and “framework” were used as terms. An example of a query is: “Agile ICT4D framework”. For the IEEE search engine the third part, method, was left out because otherwise no results would show up. Because the IEEE search engine does not automatically search for articles containing all the words in the query, ‘AND’ was used in between words to achieve that effect. For Google Scholar, if some words had to be right next to each other, they were put in between quotation marks: “Agile software development”. Then, only articles that have “Agile software development” would come back. ICT4D was also put in between quotation marks, to avoid results about ICT in

	general. No publication date restrictions were used.
9. Study selection	For the study selection the four phases of the PRISMA Statement were used: identification, screening, eligibility, and included. See Appendix A for the detailed report.
10. Data collection process	First, the eligible articles were read and any information pertaining the effect of Agile methods and ICT4D was extracted. Then, this information was categorised in one of the two ways as described in the eligibility criteria. Within each category, the information was then grouped together to form a coherent story.
20. Results of individual studies	The relevant results from the included studies can be found in chapter 3.

Table 1 – Literature review

2.5. Interview

In order to attain a greater insight into how Agile can improve ICT4D projects, an interview with ICT4D and Agile experts was held. Van Tejilingen (2014) recommends a semi-structured interview when the purpose of the interview is to elicit a person's viewpoint regarding a specific matter (p. 20). In a semi-structured interview, there are predetermined questions, but there is flexibility in asking these questions (p. 17). For example, new questions can be added ad hoc. The predetermined as well as the new questions asked during the interview can be found in Appendix B.

The interviewees are all part of W4RA, which stands for the Web alliance for Regreening in Africa. On its website, W4RA gives its mission (W4RA, n.d. -a): "to support farmer-managed regreening activities specifically by enhancing information, communication, and knowledge sharing for rural development". An example of an ICT4D project done by W4RA is RadioMarché (W4RA, n.d. -b), which is a voice-based market information system that allows farmers to advertise their products to communities in their local language. The interview was held with: prof. dr. Akkermans, who is the director; ms. drs. Bon, who is the program manager; and with ms. drs. Tuijp, who is the communication officer. When referring to their expertise, all three interviewees will be collectively referred to as "the interviewees".

3. Related work in Agile methods for ICT4D

The combination of Agile and ICT4D is rare in ICT4D research. In Google Scholar, searching for: "ICT4D Agile" returns only 387 results, as opposed to the 10200 results returned when searching for: "ICT4D" (searched for on 1-6-2017). A possible explanation can be found in the literature review by Walsham (2017), who found that the current topics within ICT4D research have to do with how ICT can contribute to development, rather than the effects of software development methods on the ICT4D projects themselves. For example, how ICT can combat systematic poverty or improve the overall health of people (p. 10). Still, work on Agile methods for ICT4D exists. In this chapter, the related work found using the method described in chapter 2.4 will be discussed. This work is divided in two categories: how Agile can affect ICT4D projects and Agile frameworks or Agile inspired frameworks for ICT4D projects.

3.1. How Agile can affect ICT4D projects

Several articles were found that discuss the benefits of using Agile methods for ICT4D projects. Bhatt, Ahmad, and Roomi (2016) analysed an ICT4D project, and found that using an Agile method allowed the developers to change the system in a natural way in response to unexpressed requirements as well as to change in the business environment (p. 34). Bon and Akkermans (2014) argue that Agile methods allow requirements to be elicited and knowledge to be created through interaction between people (p. 2). Ciaghi, Molini, Villafiorita, and Weldemariam (2013) mention that Agile methods can contribute to a healthy working environment as well as to participation of all team members (p. 8). Heimgärtner, Solanki, and Hollerit (2014) discuss the benefits of iterative development and frequent delivery on eliciting requirements, and therefore making ICT more demand-driven (p. 450). Teka, Dittrich, and Kifle (2016) also focus on how Agile can improve user involvement (p. 118). Haikin (2013) argues from experience that requirements in an ICT4D project are less clear than in a 'normal' ICT project, especially at the beginning, and that the iterative nature of Agile therefore can help (p. 6). Furthermore, Haikin argues that the stakeholders from communities from developing countries have little experience with technology, which results in these stakeholders only being able to understand their requirements by trying out the system. Because Agile methods focus on frequent delivery of software, these stakeholders would be able to try out the system sooner than when using traditional methods, thus making Agile a good fit for ICT4D projects (p. 6). According to Haikin, these two benefits combined will also result less time and money wasted on features that might later turn out to not fit the requirements. This is especially important for ICT4D projects because of the tight budgets for aid and development (p. 8).

Only one article was found that discusses a problem that might occur when using Agile methods for ICT4D projects. Haxby and Lekhi (2017) found that the individualistic, highly competitive culture in Kenya made it difficult for Kenyans to work together in teams, which is a requirement for Agile. As such, cultural barriers might hinder the learning of Agile methods.

In summary, most of the articles discussed focus on only one benefit that Agile methods can bring to ICT4D projects. This benefit is the user collaboration (e.g. improved elicitation of user requirements), which makes ICT more demand-driven. This thesis distinguishes itself by analysing the possible effects of Agile methods for more aspects of ICT4D projects.

3.2. Agile frameworks for ICT4D

Four Agile or Agile inspired frameworks specifically tailored for ICT4D projects have been found.

Bon, Akkermans, and Gordijn (2016) developed an ICT4D framework that is partially based on Agile. Other inspirations of the framework are: Living Labs, use case analysis, and requirements engineering (p. 86). The framework is specifically designed to address several ICT4D concerns, such as a lack of understanding of the local needs and the context (p. 87). Bon et al. discuss how their framework as a whole addresses the concerns they identified for ICT4D projects, and also briefly discuss the benefits of the iterative and collaborative nature of Agile in specific in the 'develop, test, & deploy' phase. Agile fosters creativity, personal commitment, and collaboration with the user (p. 98). The framework consists of five phases done iteratively (p. 88-89):

- Context analysis: understand the context and its implications. An example is understanding the circumstances in which the end users live (p. 87).
- Needs assessment: understand how ICT can add value to the end users.
- Use case and requirements analysis: understand how ICT should be designed.
- Developing, testing, and deploying of the ICT system. This is done using the Scrum method (see the Introduction for an explanation on Scrum).
- Sustainability analysis: ensure the continued service of ICT after the pilot and initial stages.

Doerflingen and Dearden (2013) developed an Agile method for commercial ICT4D projects. The method is called: "Distributed Agile Methodology Addressing Technical Ictd in Commercial Settings" (p. 1), which is abbreviated to DRAMATICS. Doerflingen en Dearden briefly discuss the effects of Agile on ICT4D projects, mentioning the benefits of Agile for collaboration with the users. The DRAMATICS method consists of three phases each with iterative cycles (p. 52-53):

1. Initial prototyping: develop a prototype for a business process.
2. Transition: install and test the prototypes as well as train the end users. The final result is deployment in the business process.
3. Productive use: support the use of the ICT in the business process.

Speedplay is a framework for ICT4D projects which takes inspiration from Agile, Action Research, and Participatory Design (Ferrario, Simm, Newman, Forshaw, & Whittle, 2014). Some of the inspirations from Agile are iterations, flexibility, and collaborative development (p. 521). Ferrario et al. argue that the Agile inspirations can benefit the user collaboration aspect of ICT4D projects (p. 521). Speedplay consists of four phases (p. 522):

- Prepare: build trust as well as elicit some early user requirements.
- Design: gain a more thorough understanding of the user requirements and design a prototype system.
- Build: collaboratively build and test the prototype agreed upon during the Design phase.
- Sustain: review work process and consider future system development.

The Nordic Model is a framework for ICT4D based on Nordic socio-cultural background and shared values, and is described as an Agile method (Hansson, Mozelius, Suhonen, Sutinen, Vesisenaho, & Wetteregn, 2009). The Nordic Model has five key aspects (p. 4-6): development is demand-driven; management is efficient, light, and non-hierarchical; design and implementation is Agile; formative evaluation is done iteratively; and equality, equity, and inclusion of users is aspired. The reasons for using an Agile method were frequent and immediate feedback from the users and informal communication to achieve equality and inclusion of all users (p. 5).

All these frameworks have been tested in ICT4D projects with success. Ciaghi, Villafiorita, and Dalvit (2014) propose that the success of Agile or Agile inspired frameworks suggests that Agile methods can be beneficial to ICT4D projects. The primary reason for using Agile methods for ICT4D seems to be improved collaboration with the user, just like in chapter 3.1. These frameworks differ in terms of other inspirations (e.g. Participatory Design for Speedplay) or application (e.g. business ICT projects for DRAMATICS).

4. Critical success factors for ICT4D projects and example projects

This chapter will seek to answer what factors are critical for a successful ICT4D project. But before that, it is essential to define what a successful ICT4D project is. Heeks (2008b) distinguishes between three different ICT4D project outcomes. The first is total failure, which occurs if projects are not implemented or are almost immediately abandoned. The second is partial failure, which occurs if major goals are not achieved, if sustainability fails, or if the project is only successful for some stakeholders. Finally, there is success, which is achieved if most major stakeholders have realized their major goals and if major undesirable outcomes have been prevented. Additionally, the project must be sustainable. The National Information Technology Forum (as cited in Jacobs & Herselman, 2005) defines sustainability as: "Being able to maintain or prolong the services with the means available" (p. 60).

In the first part of this chapter the critical success factors for ICT4D projects will be summarized. Then, three case studies performed by other researchers will be analysed to find new critical success factors and to gain a better understanding of the already found critical success factors. Finally, an overview will be given of all critical success factors.

4.1. Critical success factors for ICT4D projects

The critical success factors for ICT4D projects that will be discussed in this subsection can be found in Table 2. In the column on the left, identifiers for each critical success factors can be found, which will be used in the text to refer to the corresponding critical success factors in this table. In the column on the right, the critical success factors can be found.

Identifier	Critical success factor
ICT4D-CSF1	Monitor and evaluate regularly
ICT4D-CSF2	An ICT4D project must be demand-driven
ICT4D-CSF3	Relevant skills must be built and trained
ICT4D-CSF4	Efforts must be made to retain staff
ICT4D-CSF5	Project ownership must be given to local parties
ICT4D-CSF6	An ICT4D project must be economically self-sustainable
ICT4D-CSF7	Local partnerships must be built to achieve synergies
ICT4D-CSF8	The creation of local content must be facilitated
ICT4D-CSF9	The political context must be analysed and considered
ICT4D-CSF10	An ICT4D project must have a project champion
ICT4D-CSF11	The right technology must be chosen

Table 2 – Critical success factors for ICT4D projects from chapter 4.1

Project progress should be monitored and evaluated regularly (ICT4D-CSF1). The United Nations Development Programme (UNDP) (2001) argues that monitoring and evaluating is important for ICT4D projects, because it allows for the team members to measure the effects of ICT on development. The goal of evaluation should be to discern changes in the welfare of the members of the local community, and to adapt the project accordingly. Evaluation should be an iterative and adaptive process. According to James (2004), monitoring and evaluating also allows for problems to be identified earlier, which, if acted upon, can ensure a more effective and efficient project. Clear phases each with simple and clear objectives makes the monitoring and evaluating of projects easier. James also argues for clarity on role division and accountability, to ensure that tasks within the project are done.

In order for an ICT4D project to be successful, Ferguson and Ballantyne (2002) argue that the project must be tailored to the demands of the local community involved (ICT4D-CSF2). The ICT4D project must satisfy the present needs while also allowing for the needs of future generations. A critical element in achieving this according to Ferguson and Ballantyne is making sure that the stakeholders have ownership over the ICT4D project (ICT4D-CSF5) so as to increase their involvement in and acceptance of the ICT4D project. Mphahlele and Maepa (2003) argue the same importance of demand-driven development, saying that implementing an ICT4D project in an area where there is not sufficient demand will not result in a sustainable ICT4D project. Conradie, Morris, and Jacobs (2003) believe that the demand-driven approach will also reveal where and how technology can have the highest impact.

According to James (2004) project management, implementation, and ICT skills are scarce in developing countries. Illiteracy is also an important problem (Ciaghi & Villafiorita, 2011). If ICT4D projects are to be successful in the long run, then these skills will have to be built and trained (ICT4D-CSF3). An example of a skill that needs to be developed is technical troubleshooting (James, 2004, p. xxvi). ICT training will also help overcome technophobia (Chib & Komathi, 2009, p. 9). Ferguson and Ballantyne (2002) agree that relevant skills need to be built, but also address that it is expensive to build and train such skills. Ferguson and Ballantyne mention network-wide knowledge sharing as a possible solution to lower staff training costs. Mphahlele and Maepa (2003) argue that this training should be a continuous process focussing on essential skills specifically needed for the ICT4D project.

Aside from training staff, retaining the trained staff (ICT4D-CSF4) is equally as important because the effects of talented staff leaving can be disastrous (Ferguson & Ballantyne, 2002). Other than traditional intrinsic (e.g. praise) and extrinsic (e.g. salary) rewards, Ferguson and Ballantyne argue that ownership (ICT4D-CSF5) can be a major motivational incentive due to the involvement it brings. Knowledge sharing may reduce the harmful effects of trained staff leaving.

According to Ferguson and Ballantyne (2002), local ownership of the project is important for the long-term success of ICT4D projects (ICT4D-CSF5). Ferguson and Ballantyne define local ownership in the following way: “Local development actor or actors takes responsibility for the conceptualisation, formulation and implementation of an activity, or parts of it” (p. 4). So ownership is not just about owning the ICT, but also about defining the objectives of the ICT4D project, and contributing towards results. Local ownership is related to two others critical success factors: it improves the alignment of the ICT4D project to the needs of its stakeholders (ICT4D-CSF2) and it improves the motivation of staff (ICT4D-CSF4). Mphahlele and Maepa (2003) argue that successful local ownership will result in the community viewing the ICT as an integral part of their daily lives.

According to Butcher (as cited in Mphahlele & Maepa, 2003), many ICT4D projects rely on donor money for their continued survival. Butcher argues that such ICT4D projects are frail due to the risk of donors pulling out. An ICT4D project should thus be economically self-sustainable (ICT4D-CSF6). Keniston and Kumar (2003) argue for economic self-sustainability as well, and noticed projects falling apart as soon as enthusiasm and funding from outside partners disappears. Keniston and Kumar however do argue for the importance of donor money, stating that due to the experimental nature of many ICT4D projects it cannot be expected for them to be profitable from the get-go. But plans for long-term economic sustainability have to be made for after the experimental phase of an ICT4D project passes. Bon et al. (2016) also argue that donor money is necessary in order to set up an ICT4D project, and that plans for economic self-sustainability have to be made (p. 88). Mphahlele and Maepa proposes that an ICT4D project should be set up as a business venture, meaning that an ICT4D project should be able to generate income and profit (ICT4D-CSF6). This will also allow the community to understand better the cost implications of using ICT. One important dimension to economic self-sustainability, as Mphahlele and Maepa identified, is that the business venture should

apply a competitive pricing strategy. This becomes especially important in the face of competitors. Another important dimension is the marketing strategy, because Mphahlele and Maepa noticed that one of the main reasons why ICT4D projects fail, is because of the inability to inform the community about the benefits of ICT4D projects.

Ferguson and Ballantyne (2002) argue for the importance of building local partnerships (ICT4D-CSF7). A network of local partnerships will allow for the participants to gain access to resources they might otherwise not have had access to, such as skilled people or financial mechanisms (p. 12). The interaction between local partners will also give insight in their wishes, which if capitalized upon might decrease stakeholder turnover. The UNDP (2001) supports the importance of exploiting such synergies as well. Mphahlele and Maepa (2003) also argue for local partnerships, saying that complementary services or products can increase the marketability and profitability of ICT4D projects.

The ability of locals to develop local content (ICT4D-CSF8) plays an important role in their use of ICT (Keniston & Kumar, 2003). Keniston and Kumar describe local content as content being in local language as well as having inspiration from local culture, created by locals. The reason for the importance of local content is that only a select portion of the population will be able to understand content from, for example, The United Kingdom, due to language and cultural barriers. Keniston and Kumar estimate that in South Asia alone 95% of the population would not be able to access and understand non-local content. An example of what local content can be is information for farmers regarding which vegetables can be grown on their fields (Conradie et al., 2003, p. 215).

Ferguson and Ballantyne (2002) argue that the political situation in a country can affect an ICT4D project on two levels: micro and macro level. On a micro level issues regarding ownership can arise due to a lack of defined ownership over processes and resources, or from unsuccessful transfers of ownership. On a macro level issues can arise due to increased bureaucracy or because the project is turned into a political statement. On both levels the progress can be hindered and thus Ferguson and Ballantyne argue that it is imperative to gain an understanding of the political context in order to minimize the effects (ICT4D-CSF9). The UNDP (2001) argues that information by itself already attracts political attention, and may limit the effectiveness of projects. The UNDP mentions restricted access to Internet as an example (p. 6).

Renken and Heeks (2013) argue that every project should have an ICT4D project champion in order to be successful (ICT4D-CSF10), which they define as follows:

> "Any individual who makes a decisive contribution to the ICT4D project by actively and enthusiastically promoting its progress through critical stages in order to mobilise resource and/or active support and cooperation from project stakeholders" (p. 129)

This recommendation is shared by James (2004), however James warns about over reliance on a few individuals. If these individuals were to be removed from the project, James suspects that the projects might easily collapse. Thus, for projects to be sustainable in the long run multiple ICT4D champions are necessary. Mphahlele and Maepa (2003) note that a project champion from the local community can help with establishing and safeguarding community support and ownership.

Ferguson and Ballantyne (2002) argue that the technology chosen plays an important role in the long-term success of ICT4D projects (ICT4D-CSF11). The reliability of ICT infrastructure, the availability of technology, and the maintenance and upgrading of ICT are mentioned by Ferguson and Ballantyne as key factors. Technology also needs to be affordable for the people involved with the ICT4D project (James, 2004). The UNDP (2001) argue that the right technology must be chosen

for the task, mentioning that a simple system can at times be far more effective than a state-of-the-art system.

In summary, a total of eleven critical success factors for ICT4D have been identified thus far. Differences in occurrences of these critical success factors in literature will be discussed in chapter 4.3. As discussed in the Research Approach (more specifically, chapter 2.1 and 2.3), these critical success factors for ICT4D projects will be used to determine if Agile can work in ICT4D projects, and to determine in what ways Agile can improve the success of ICT4D projects.

4.2. Examples of ICT4D projects

In this subsection three ICT4D projects will be summarized and analysed. The purpose of doing this, as well as the selection criteria for the three ICT4D projects, can be found in chapter 2, the Research Approach.

4.2.1. The Happy Valley Project

Krauss (2016) analysed an ICT training project called The Happy Valley Project. Happy Valley is the name of a rural town in South Africa. One key characteristic of the Happy Valley community is that it has strong cultural practices and values. Furthermore, Happy Valley is described as being among the most economically disadvantaged towns in South Africa. The goal of the project was to train teachers according to the UNESCO ICT Competency Framework for Teachers (2011). The framework postulates that teachers should not only have ICT competencies and be able to transfer those to students, but also promote a collaborative and creative use of ICT. The project analysis by Krauss thus fits the critical success factor of building and training relevant skills (ICT4D-CSF3), though in this project that building and training of skills is also the primary objective.

The project started with a pilot training initiative (Krauss, 2016). The pilot consisted of computer literacy training for the teachers and other important caregivers. The training required literacy in writing and reading as well as the ability to speak English. For those who did not meet the aforementioned requirements, a course dealing with pre-basic ICT needs was given. The training sessions were scheduled from morning until 15:00, but often lasted until late in the evenings due to the low skill level of teachers as well as their need for personal interaction.

Concerns before the training started were that the teachers might not fully appreciate the importance and value of the training, as well as possibility of truancy due to the training being free (Krauss, 2016). Additionally, there were some trust issues due to past experiences with outsiders trying to profit from the community. Due to the project leaders realizing these concerns and taking steps to remedy them early on, the training started with plenty of commitment and dedication from the teachers. The greatest challenge during the project was the difference in culture. Krauss describes his experience regarding the difference in culture as follows: “My interactions with the community and my feeble efforts to gain their trust were constantly challenged by my consistent misunderstandings of their different cultural world” (p. 20). The presence of culture interpreters helped in such difficult situations.

The value of this project analysis by Krauss (2016) for this thesis lies in the need to inspire and motivate the community (in this case the teachers), as otherwise training will not be effective or might be skipped upon. One important way to achieve this was ownership by a local and the inclusion of that owner (p. 21) (ICT4D-CSF5). Additionally, cultural barriers need to be overcome and trust needs to be built in order for communication to go well. At one point during the project this

failed, and as a result some teachers were traumatised and left with computer phobia. One teacher did not recover by the time the project finished (p. 23).

In regards to the other critical success factors identified in chapter 4.1, Krauss (2016) did not highlight their importance, but did discuss some of them. In setting the project up key stakeholders from the community were included to ensure a demand-driven approach (p. 17) (ICT4D-CSF2). The project relied on donor money (p. 10), but economic self-sustainability was not mentioned (ICT4D-CSF6). There were multiple project champions, both from inside and outside the community (p. 13) (ICT4D-CSF10). Before the start of the project the choice of technology was discussed with special attention to the school's budget (p. 19) (ICT4D-CSF11).

4.2.2. The Sichuan rural informatization program

Liu (2016) performed a case study on the Sichuan rural informatization program. Sichuan is a province in China and the program started in 2001. The goal of the program was to improve the telecommunication connectivity (e.g. telephone network, broadband) and content (e.g. local news) for rural areas. In China, connectivity and content were historically offered by different parties: telecommunication carriers were in charge of creating connectivity and the government was in charge of creating content. So whereas there existed political pressure due to the program to improve the connectivity of rural areas, ultimately the telecommunication carriers had to provide that. But to the telecommunication carriers, providing more connectivity was not economically viable due to low demand. The demand for connectivity in rural areas could increase by providing more content, however the government was in charge of that. Finally, the government did not use the network created by the telecommunication carriers, but rather their own, poor network. There existed a synergy between the two services, and as such, as part of the program, in 2003 strategic partnerships were formed between connectivity providers and content creators. Content owners would provide their content to the connectivity providers, which would give the connectivity providers more incentive to provide connectivity.

Liu (2016) proceeded with discussing the sustainability of the project. China's policy making design implies that the central government cannot issue binding orders to provincial governments (and vice versa). For this program that meant the Sichuan provincial government could not order the state-owned carriers to provide more connectivity. Furthermore, due to a lack of revenue-sharing plan, content creators were not incentivized to share their content with the carriers. Together, these two problems resulted in the strategic partnerships being of a more symbolic value. In practice, the financial burden of informatization (including the creation of content) lied primarily on the telecommunication carriers. The carriers were willing to do so however, because a content-rich informatization service was considered an opportunity to attract and retain rural customers. It was also expected that the fees charged would cover the cost, thereby making the project economically self-sustainable. The carriers applied a supply-driven approach to content delivery. Although Liu admitted to only interviewing a few people, most of the people he interviewed did not perceive the content delivered as relevant (p. 8). But the profit margins were small or non-existent so eventually the carriers shifted their focus to more profitable areas, such as urban areas. The telecommunication carriers are listed on domestic and foreign stock markets, and this shareholder ownership was listed as a reason for shifting the focus to more profitable areas (p. 7). The carriers were not able to pull out completely due to political pressure, so instead the informalization has become stagnant.

The Sichuan informatization program (Liu, 2016) is relevant to this thesis in several ways. It showcases the importance of economic self-sustainability (ICT4D-CSF6), while at the same time showing a darker side that might surface if economic self-sustainability is the primary driver: areas that are not profitable might be ignored. It also discusses the importance of understanding the

political context, and policy making in particular (ICT4D-CSF9). The policy making design discussed in the previous paragraph could have been potentially solved by a higher focus on assigning responsibilities (p. 9). Furthermore, the program provides insight in how political interests could positively influence a project. It is the political interests that restrain the telecommunication carriers from divesting the program, and it is the political interests that could force the carriers to improve the program. Finally, the program highlights the importance of local content (ICT4D-CSF8) and Liu highlights the importance of demand-driven development (ICT4D-CSF2), both of which are discussed as ways to improve the economic self-sustainability.

4.2.3. Impact Venture

Impact Venture is an Information Technology Business Process Outsourcing (IT-BPO) for-profit social enterprise (Sandeep & Ravishankar, 2016). Impact Venture is active in several industries, such as e-commerce, web services, and insurance, and uses a part of their profits towards the development of local communities in the Himalayas (within India). Impact Venture has a partnership with the local non-profit organization Rural Lives. Rural Lives supports Impact Venture by improving their relationship with the local communities, as well as by managing operations and hiring and training new recruits. Impact Venture has set up an advisory committee in all five villages in which they are active, which debates the plans and operations of Impact Venture and any problems that occur. This advisory committee consists of representatives of both Impact Venture and Rural Lives, and the leaders of the local communities.

The objective of the paper was to understand the social and psychological factors that are of importance for impact sourcing businesses (Sandeep & Ravishankar, 2016, p. 1). Impact sourcing is defined as outsourcing work (in this case, ICT-related work) to the poor. Sandeep and Ravishankar achieved the objective by looking at four themes: bringing progress to the local community, positioning of the company within the local community, providing material benefits for the local community, and providing equal opportunity to the local community (p. 14). Impact Venture brought progress to the local communities through job creation, thereby remedying the most important problems facing the communities: “scarcity of jobs and an uncertain future for the youth” (p. 19). Impact Venture positioned itself within the local communities as family, through intensive interaction with said local communities. Initially, Impact Venture faced much opposition because they had no foundation within the local communities. After trying to understand the culture of local communities better, Impact Venture realised the value of family was incredibly important. Positioning themselves as family thus helped Impact Venture to become accepted within the local communities. The conflict of culture was also important in how Impact Venture brought liberal values to the local communities, such as boys and girls working together (p. 23). The older members in particular resisted these values (p. 22). By relating these liberal values to material benefits such as job creation, Impact Venture managed to change the local communities’ perception of them. Instead of bringing undesirable cultural change, Impact Venture is now seen as a benefactor (p. 16). Finally, Impact Venture has to constantly make sure the local communities perceive their efforts as fair. In one instance, villagers who failed a job interview were quick to judge Impact Venture as being partial, and vandalized an Impact Venture centre as a result (p. 25). Impact Venture attempts to remedy this through, for example, constant communication of impartialness and by providing training for everyone who is willing (p. 26).

Impact Venture is a good example of the importance of demand-driven ICT4D projects (ICT4D-CSF2). By focussing on resolving the problems that were of importance to the local communities, Impact Venture was able to gain the trust and commitment of said local communities (Sandeep & Ravishankar, 2016). Furthermore, the issue of understanding the culture is raised in order to prevent or resolve conflicts arising from cultural differences. Forming a local partnership (ICT4D-CSF7) with

Rural Lives also benefitted Impact Venture greatly in the ways described in the first paragraph of this section. Finally, this case highlights that understanding the local political context (ICT4D-CSF9) is of importance, because it allowed Impact Venture to realize that the resistance from the leaders was because of the cultural differences and lack of familiarity.

4.2.4. Findings from the ICT4D projects

By looking at the ICT4D projects together, two new critical success factors can be identified: developing a cultural understanding and building trust. The critical success factors for ICT4D projects that will be discussed in this subsection can be found in Table 3. In the column on the left, identifiers for each critical success factors can be found, which will be used in the text to refer to the corresponding critical success factors in this table. In the column on the right, the critical success factors can be found.

Identifier	Critical success factor
ICT4D-CSF12	Develop a cultural understanding of the local community
ICT4D-CSF13	Build trust between the local community and outside parties

Table 3 – Critical success factors for ICT4D projects from chapter 4.2.4

From The Happy Valley Project (Krauss, 2016) and Impact Venture (Sandeep & Ravishankar, 2016) it becomes clear that cultural understanding is a critical success factor (ICT4D-CSF12). For The Happy Valley Project cultural understanding was necessary to avoid conflicts during the constant interaction between outsiders and the local community (Krauss, 2016, p. 20). Krauss suggests to include cultural interpreters in the project (p. 25). For Impact Venture cultural understanding was necessary to become accepted within the local community and gain their trust, as well as to gain access to their resources (Sandeep & Ravishankar, 2016, p. 20). Impact Venture managed to overcome the cultural barrier due to a local partnership (p. 17-18) and by having a strong presence within the local community (p. 21).

Mayer, James, and Schoorman (1995) define trust as follows:

> "the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other part" (p. 712)

Trust can be a contributing factor to the willingness to cooperate with another party, and becomes necessary if that cooperation results in the trustor being put at risk (Mayer, James, & Schoorman, 1995, p. 712). In an ICT4D project the local community would be the trustor, and the outside party the trustee. Examples of risk in the context of ICT4D projects are outside parties attempting to take advantage of local communities (Krauss, 2016, p. 19), and change to cultural values (Sandeep & Ravishankar, 2016, p. 21). For both The Happy Valley project (Krauss) and Impact Venture (Sandeep & Ravishankar), an initial lack of trust hindered cooperation.

Trust between the local community and the outside party thus needs to be built (ICT4D-CSF13). Mayer, James, and Schoorman (1995) identify two factors that determine the level of trust: the trustor's propensity to trust and the trustee's perceived trustworthiness (p. 715). The propensity to trust differs among individuals, but factors that influence the propensity are history with development, personality, and the culture (p. 715). Trustworthiness has ability, benevolence, and integrity as antecedents (p. 715). Ability refers to the skills and expertise of a party within a domain (p. 717). For ICT4D projects, this can be knowledge about ICT or business strategy for example (ICT4D-CSF3). Benevolence refers to what degree the party desires to help the trustor without

regard to extrinsic rewards (p. 718). Finally, integrity refers the trustee's adherence to principles. In terms of perceived trustworthiness, these principles must be of importance to the trustor (p. 719). Perceived trustworthiness is a dynamic factor that changes due to the results from trust-taking behaviour (p. 728).

Interesting is how the different critical success factors have a different importance in different projects, sometimes even appearing to be irrelevant. For example, creating local content (ICT4D-CSF8) was of critical importance for the Sichuan rural informatization program (Liu, 2016), but appears to have been irrelevant for Impact Venture (Sandeep & Ravishankar, 2016). Another example is that understanding the political context (ICT4D-CSF9) was of critical importance for the Sichuan rural informatization program and for Impact Venture, but appears to have been irrelevant for The Happy Valley Project (Krauss, 2016). It could be that the critical success factors are not always of critical importance for all ICT4D projects. An alternative possibility is that the critical success factors are always of critical importance for all ICT4D projects, but that some were simply not discussed in the case studies from the other researchers. This is plausible because the other researcher approached the case studies with a certain goal in mind. For example, Sandeep and Ravishankar (2016) analysed the Sichuan rural informatization program to judge the sustainability of the program. This observation indicates more research is necessary to determine the occurrence of critical success factors. This will be discussed in the Further Research chapter.

4.3. Overview and occurrence of the critical success factors

An overview of all the critical success factors for ICT4D projects discussed in this chapter can be found in Table 4. In the column on the left, identifiers for each critical success factors can be found. In the column in the middle, the critical success factors can be found. Finally, in the column on the right the number of occurrences in the selected literature is mentioned.

Identifier	Critical success factor	Number of occurrences
ICT4D-CSF1	Monitor and evaluate regularly	3
ICT4D-CSF2	An ICT4D project must be demand-driven	5
ICT4D-CSF3	Relevant skills must be built and trained	6
ICT4D-CSF4	Efforts must be made to retain staff	2
ICT4D-CSF5	Project ownership must be given to local parties	3
ICT4D-CSF6	An ICT4D project must be economically self-sustainable	7
ICT4D-CSF7	Local partnerships must be built to achieve synergies	4
ICT4D-CSF8	The creation of local content must be facilitated	5
ICT4D-CSF9	The political context must be analysed and considered	5
ICT4D-CSF10	An ICT4D project must have a project champion	3
ICT4D-CSF11	The right technology must be chosen	5
ICT4D-CSF12	Develop a cultural understanding of the local community	1
ICT4D-CSF13	Build trust between the local community and outside parties	0

Table 4 – Critical success factors for ICT4D projects from chapter 4

To determine the occurrence of the critical success factors for ICT4D projects the literature referred to in this chapter was used. However, to not distort the picture created by these occurrences, only literature that sought to create a list of critical success factors was used. So for example, The Happy Valley project (Krauss, 2016) was not used to determine the occurrence of critical success factors, because the author sought to specifically analyse cultural and power issues. The occurrence of critical success factors for ICT4D projects for each selected piece of literature can be found in Table 5. In the first column the identifiers for the critical success factors are written. The other columns

specify whether that critical success factor is mentioned in the article specified in the top cell of that column.

Identifier	UNDP (2001)	Ferguson and Ballantyne (2002)	Conradie, Morris, and Jacobs (2003)	Keniston and Kumar (2003)	Mphahlele and Maepa (2003)	James (2004)	Touray, Salminen, and Mursu (2013)
ICT4D-CSF1	Yes				Yes	Yes	
ICT4D-CSF2	Yes	Yes	Yes	Yes	Yes		
ICT4D-CSF3	Yes	Yes	Yes		Yes	Yes	Yes
ICT4D-CSF4		Yes				Yes	
ICT4D-CSF5	Yes	Yes			Yes		
ICT4D-CSF6	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ICT4D-CSF7	Yes	Yes			Yes	Yes	
ICT4D-CSF8	Yes		Yes	Yes		Yes	Yes
ICT4D-CSF9	Yes	Yes		Yes		Yes	Yes
ICT4D-CSF10	Yes				Yes	Yes	
ICT4D-CSF11	Yes	Yes	Yes			Yes	Yes
ICT4D-CSF12							Yes
ICT4D-CSF13							

Table 5 – Stability test for the critical success factors for ICT4D projects from chapter 4

Based on the number of occurrences, the most important critical success factor is that a project must be economically self-sustainable (ICT4D-CSF6), followed by the building and training of the relevant skills (ICT4D-CSF3). Four other critical success factors that have a high number of occurrences are: a project must be demand-driven (ICT4D-CSF2), the creation of local content must be facilitated (ICT4D-CSF8), the political situation must be analysed and considered (ICT4D-CSF9), and the right technology must be chosen (ICT4D-CSF11). The low number or lack of occurrences for developing a cultural understanding (ICT4D-CSF12) and building trust (ICT4D-CSF13) is explained by the absence of the discussed projects in chapter 4.2. Finally, it should be noted that all critical success factors are of importance to the success of a project. It could be possible that critical success factors with a low number of occurrences are underrepresented in literature (see chapter 8.2 for the implications).

5. Agile software development

This chapter will provide a more in depth analysis of Agile than was done in the Introduction. The critical success factors for Agile will be summarized in order to understand in what situations Agile is applicable, as well as to provide a more thorough understanding of how Agile works. Then, advantages of Agile will be summarized. Finally, the limitations of Agile will be summarized.

5.1. Critical success factors for Agile software development

The critical success factors for Agile that will be discussed in this subsection can be found in Table 6. In the column on the left, identifiers for each critical success factors can be found, which will be used in the text to refer to the corresponding critical success factors in this table. In the column on the right, the critical success factors can be found.

Identifier	Critical success factor
Agile-CSF1	Team environment
Agile-CSF2	Team capability
Agile-CSF3	User involvement
Agile-CSF4	Project management
Agile-CSF5	Agile software engineering techniques
Agile-CSF6	Delivery strategy

Table 6 – Critical success factors for Agile software development from chapter 5.1

Chow and Cao (2007) performed a survey study to the critical success factors of Agile projects and tested several potential success factors on four dimensions to project success. These four dimensions are quality (the quality of the delivered product), scope (to what degree the product meets the user's requirements), timeliness (whether the product is delivered on time or not), and cost (whether the real costs and effort put in were as projected). Chow and Cao found six critical success factors: team environment, team capability, user involvement, project management, Agile software engineering techniques, and delivery strategy.

A good team environment contributes positively to the quality of the product (Chow & Cao, 2007) (Agile-CSF1). In a good team environment the entire team should be located in a single place, the team should be small, and the team should be self-organizing. If a project has multiple teams they should work collaboratively rather than independently. Teams should be small because as a team has more members, coordination becomes more difficult (Lindvall et al., 2002; Turk, France, & Rumpe, 2014). Mahanti (2006) also states that the larger a team grows, the more difficult it is to work in the same physical location.

Team capability is another critical success factor and was found to positively contribute to the timeliness and cost of a project (Chow & Cao, 2007) (Agile-CSF2). A good team member should have high competence, expertise, and motivation. A good manager should have an adaptive management style and possess knowledge on Agile. Finally, relevant technical training should be provided to the team members. Lindvall et al. (2002) argue for the importance of having competent team members, stating that because smaller teams for Agile are preferable, individual skill levels must be higher. According to Mahanti (2006), the team should pay meticulous attention to the Agile principles, as otherwise the coveted results will not be achieved. Hilkka, Tuure, and Rossi (2005) argue that developers must also possess domain knowledge in order to be able to communicate with the user.

User involvement was found to contribute positively to the scope of the product (Chow & Cao, 2007) (Agile-CSF3). To achieve good user involvement, a positive user relationship should be built. The user should have complete authority regarding the project. Finally, the user should have a strong commitment and presence. Lindvall et al. (2002) argue that because Agile is based on close interaction with users, the faster users deliver feedback the better.

Project management processes contribute positively to the quality of the product (Chow & Cao, 2007) (Agile-CSF4). Requirement management processes, project management processes, and configuration management processes should all be Agile. A working schedule should be put in place and followed. Progress should be tracked. There should be a strong focus on communication, for example with daily face-to-face meetings. According to Misra, Kumar, and Kumar (2009), the corporate culture should be supportive of Agile practices.

Agile software engineering techniques were found to contribute positively to the quality and scope of the product (Chow & Cao, 2007) (Agile-CSF5). According to Chow and Cao these techniques are: coding standards, simple design, refactoring, limited but sufficient documentation, and integration testing. Refactoring is making changes to the code in order to improve it, but without changing the behaviour of the code (Fowler, 1999).

Finally, the delivery strategy positively contributes to the scope of the product, and the timeliness and cost of the project (Chow & Cao, 2007) (Agile-CSF6). A good delivery strategy prioritizes the important features of the product first. Furthermore, software should be regularly delivered.

In summary, a total of six critical success factors for Agile have been identified. As discussed in the Research Approach (more specifically, chapter 2.1 and 2.3), these critical success factors for Agile will be used to determine if Agile can work in ICT4D projects.

5.2. Advantages of Agile software development

The advantages of Agile that will be discussed in this subsection can be found in Table 7. In the column on the left, identifiers for each advantage can be found, which will be used in the text to refer to the corresponding advantage in this table. In the column on the right, the advantages can be found.

Identifier	Advantage
Agile-ADV1	More robust to changing requirements
Agile-ADV2	Improved communication with the user
Agile-ADV3	Higher quality of software
Agile-ADV4	Increased user satisfaction
Agile-ADV5	Good, internal communication
Agile-ADV6	Improved employee job satisfaction
Agile-ADV7	A higher return on investment
Agile-ADV8	Increase in successful projects
Agile-ADV9	Improved control over projects
Agile-ADV10	Improved organizational learning

Table 7 – Advantages of Agile software development from chapter 5.2

Sillitti, Ceschi, Russo, and Succi (2005) argue that requirements are inherently variable because both the developer and user acquire more knowledge about the domain of the application (p. 21). Requirements also change because the business environment in which the user is positioned is dynamic (Cao & Ramesh, 2008, p. 60). Adaptability is thus an important characteristic that a

software development method must possess. Sillitti et al. performed a survey to determine the most important problem encountered in development for companies using Agile methods and companies using traditional methods (p. 21). 88% of the companies using traditional methods reported changing requirements as the most important problem, whereas only 12% of the Agile companies did the same. Two reasons explain this difference (Sillitti et al., p. 22). First, Agile firms typically use a more simple software architecture, postponing any complex and binding changes as much as possible. This makes the architecture more robust to change. Second, Agile firms typically allow for requirements variability in the contract between the developer and the user. In those situations, users can specify or adjust requirements at the beginning of each iteration. Agile methods are thus more robust to changing requirements (Agile-ADV1).

In Agile methods, face-to-face communication with the user instead of rigorous documentation is the norm (Cao & Ramesh, 2008, p. 63). Because documentation can be time-consuming, this aspect of Agile saves time. The iterative nature of Agile allows for more frequent communication with the user (p. 64). Furthermore, this face-to-face-communication allows for an improved elicitation of requirements, which reduces the likelihood of requirements changing later on (Cao & Ramesh, p. 63; Pikkarainen, Haikara, Salo, Abrahamsson, & Still, 2008, p. 321). Agile methods seek to validate the requirements repeatedly starting from the beginning, which also reduces the likelihood of requirements changing (Cao & Ramesh, p. 65; Karlström & Runeson, 2005, p. 46). Agile methods thus improve the communication between the developer and the user by making the communication more efficient, frequent, and purposeful (Agile-ADV2).

In his analysis of attention to software quality within Agile methods, Ambler (2005) argues that software quality principles quality professionals have been preaching for are included in Agile methods, thus resulting in higher quality software (Agile-ADV3). An example of such a principle is test driven development, which is an approach that suggests writing automated tests first, and code afterwards if the tests fail (Janzen & Saiedian, 2005). This is done iteratively. Hayes' study (as cited in Mahanti, 2006) came to the same conclusion, but mentioned frequent user feedback as another contributing factor.

Agile methods improve user satisfaction (Agile-ADV4). An important contributing factor is the improved communication with users (Agile-ADV2) (Ilieva, Ivanov, & Stefanova, 2004, p. 7; Mann & Maurer, 2005, p. 77). Another important contributing factor is the increased user involvement in Agile methods (Ceschi et al., 2005, p. 24; Mann & Maurer, 2005, p. 77), which is a critical success factor for Agile methods (Agile-CSF3). Finally, the improved quality of software (Agile-ADV3) increases user satisfaction (Mann & Maurer, 2005, p. 77).

Agile methods should have a strong focus on internal communication (Agile-CSF5). In Scrum for example, meetings are held multiple times a week, usually daily (Rising & Janoff, 2000, p. 30). This frequent communication results in an improved understanding of the requirements, tasks, project status, and resource allocation among all team members (Pikkarainen et al., 2008, p. 316-317, Chong, 2005, p. 46). Hayes (as cited in Mahanti, 2006) argues that because of a higher focus on teamwork in Agile, there is more flexibility in projects due to less dependency on individuals. For example, the possibility of bottlenecks is eliminated because developers no longer work individually (p. 197). Agile methods thus have good, internal communication (Agile-ADV5).

Mannaro, Melis, and Marchesi performed a survey on job satisfaction among employees of traditional and Agile methods (p. 171-172), and found that the job satisfaction is higher for employees of Agile methods (Agile-ADV6). Six reasons were given to indicate this higher job satisfaction: employees experienced less stress, felt more productive, enjoyed the internal communication (Agile-ADV5), found the job environment more pleasant and comfortable, were

more motivated, and were more willing to continue using their software development method. Mann & Maurer (2005) found that the improved software quality (Agile-ADV3) also contributed to a higher job satisfaction.

Rico (2008) found that the return of investment (ROI) is higher in projects done with Agile methods (Agile-ADV7). Rico attributes this to numerous factors: “increased cost-effectiveness, productivity, quality, cycle-time reduction, and customer satisfaction” (p. 6). Hayes’ study (as cited in Mahanti, 2006) attributes the higher ROI simply to the iterative nature of Agile. Hayes argues that the iterative cycles allow for more frequent user feedback, which in turn leads to better software and thus a higher ROI.

Another advantage is that Agile reduces the likelihood of a project failing (Agile-ADV8). Serrador and Pinto (2015) performed a survey on project success across multiple industries and countries and found that as more of the Agile approach was applied in the project, the higher the reported project success was. Hayes’ study (as cited in Mahanti, 2006) attributed this occurrence again to the iterative nature of Agile. The iterative cycle of performing the abstract tasks (analysis, design, implementation, testing, and maintenance) as opposed to a sequential approach of these tasks allow for more visibility of the project. With this increased visibility the potential success of the project would then become clearer, which gives insight into whether adjustments can or have to be made, or if the project has to be cancelled entirely.

Hayes (as cited in Mahanti, 2006) argues that Agile methods improve the control over projects (Agile-ADV9). Multiple factors are mentioned that contribute to this: “Short iterations, multi-disciplinary teams, knowledge sharing, continuous integration, and feedback” (p. 197). Karlström and Runeson (2005) argue that the sense of control comes the subdividing of tasks. This would allow team members to focus more and experience less confusion. An example of this improved control over projects is that important decisions can be made quicker, thus allowing Agile teams to respond to emergent situations quicker (Misra et al., 2009).

Agile methods improve organizational learning (Agile-ADV10). Nerur, Mahapatra, and Mangalaraj (2005) argue that Agile focusses on teamwork and foster organizational learning within those team (p. 76). Holz and Maurer (2002) believe that Agile methods have some potential for knowledge sharing due to pair programming (p. 61), which Williams, Kessler, Cunningham, and Jeffries (2000) define as a practice where two programmers work collaboratively on one computer (p. 19). Tessem (2003) argues that pair programming improves learning (p. 133). However, knowledge sharing might also be difficult when using Agile methods. According to Cockburn (as cited in Holz & Maurer, 2002), knowledge is more implicit than explicit when using Agile methods, because documentation is kept to a minimum and communication occurs informally. This can result in several problems relating to knowledge sharing (Holz & Maurer, 2002, p. 60-61): similar questions may keep being asked, solutions to problems that have been faced before may be forgotten, knowledge sharing between different teams may not occur, and knowledge may leave the organization when developers leave. But Nerur et al. (2005) propose that agreements could be made to make certain knowledge explicit (p. 76).

In summary, a total of ten advantages of Agile have been identified. As discussed in the Research Approach (more specifically, chapter 2.1 and 2.3), these advantages of Agile will be used to determine in what ways Agile can improve ICT4D projects.

5.3. Limitations of Agile software development

Identifier	Critical success factor
Agile-LIM1	Does not work well for distributed development
Agile-LIM2	Does not work well for developing large, complex systems

Table 8 – Limitations of Agile software development from chapter 5.3

The critical success factors discussed in chapter 5.1 limit the possible scenarios in which Agile methods can be used. For example, for software development involving large developer teams Agile is not a good approach (Agile-CSF1). Another example are software development projects where user involvement are not viable (Agile-CSF3). In this subsection however, such limitations are not covered.

Turk et al. (2014) argue that Agile has limited support for distributed development environments (Agile-LIM1). Distributed development is a form of development, in which development occurs in multiple locations spread across the globe (Sengupta, Chandra, & Sinha, 2006, p. 731). Ramesh, Cao, Mohan, and Xu (2006) have identified several challenges that occur when using Agile methods in distributed development environments (p. 42). Ramesh et al. argue that the geographical distances result in a more formal way of developing: distributed development relies on documentation as the main communication channel, on fixed requirements, on control through processes, and on contracts. Agile on the other hand, relies on informal communication channels (such as daily meetings), on changing requirements, on control in the hands of developers, and on informal agreements. Furthermore, Ramesh et al. argue that the geographical distance prevents from team cohesion to form, which is an important aspect of Agile. Turk et al. state that the distance between development teams and the user hinders the value of interaction. While technologies exist to remedy the difficulty of communication, Turk et al. argue that these technologies are not only expensive, but also less effective than face to face communication. Mahanti (2006) also discusses this limitation and stresses the difficulty of integrating development teams that are separated by geographical lines.

Dybå and Dingsøyr (2008) found in their systematic literature review on Agile that Agile methods pay less attention to software architecture than traditional methods. This limits the compatibility of Agile methods for the development of large, complex systems (Agile-LIM2) (Blake, 2010), because architecture is important for such systems (Garland & Shaw, 1993, p. 2; Turk et al., 2014, p. 45). For example, Agile methods avoid making big architecture decisions up front (Mancl, Fraser, Opdyke, Hadar, & Hadar, 2009, p. 1). For large, complex systems however, it may be necessary to design for important aspects that will be costly to change later (Turk et al., 2014).

In summary, a total of two limitations of Agile have been identified. As discussed in the Research Approach (more specifically, chapter 2.1 and 2.3), these limitations of Agile will be used to determine if Agile can work in ICT4D projects.

6. Agile software development and ICT4D combined

This chapter will seek to answer how suitable Agile is for ICT4D projects. First will be determined if Agile can work for ICT4D projects by comparing the critical success factors for both. Then, by combining the critical success factors for ICT4D projects and the advantages and characteristics for Agile, the ways in which Agile can improve ICT4D projects will be determined.

6.1. Determining if Agile software development can work in ICT4D projects

For each critical success factor for Agile will be determined if they can be satisfied at the beginning of an ICT4D project, or if they can only be satisfied in later stages of an ICT4D project (in other words: after satisfying certain critical success factors for ICT4D projects).

The team environment (Agile-CSF1) has some aspects that are also important for ICT4D projects. For ICT4D projects it is beneficial for all the team members to be located near each other. This is because an active presence within the local community can contribute to developing a cultural understanding (ICT4D-CSF12), and because ownership over the project by the local community requires their active participation (ICT4D-CSF5). Prakash and De' (2007) also argue for ICT4D projects to be done on location to allow for better monitoring of the factors influencing the ICT4D projects. However, this is not always possible. For example, according to the interviewees (see chapter 2.5; Appendix B), members of W4RA go several times a year for extended periods. The goal is to do as much work as possible in those time frames. For example, by creating prototypes overnight.

One aspect of the team environment will be more difficult to satisfy, however. That is the requirement that teams should be small to work effectively. According to Thapa and Hatakka (2017), ICT4D is a multidisciplinary field that often includes theories from the following fields: "Information systems, development studies, business and political sciences" (p. 2579). For example, in one ICT4D project, people from social sciences, ICT, and art sciences were included (Shiang, Halin, Lu, & CheeWhye, 2016). Specific roles can be identified based on the critical success factors for ICT4D discussed in chapter 4, for example trainer (ICT4D-CSF3) and cultural interpreter (ICT4D-CSF12). Dearden, Rizvi, and Gupta (2010) suggest the role of Development Project Co-ordinator, who, amongst other tasks, is in charge of seeking ways in which technology can be used to bring benefits in the local community in collaboration with the local community (p. 9). The complexities of ICT4D projects thus result in more roles to be assigned, which may complicate reaching the ideal size of teams. In Scrum for example, the team size should not exceed ten team members (Rising & Janoff, 2000, p. 27). With larger teams, coordination and working on location becomes more challenging.

User involvement (Agile-CSF3) requires cooperation with the local community, which is also important for ICT4D projects (ICT4D-CSF2). From chapter 4, important prerequisites for cooperation can be identified: developing a cultural understanding (ICT4D-CSF12) and building trust (ICT4D-CSF13). Multiple factors contribute to developing a cultural understanding. These factors are cultural interpreters, local partnerships, and a strong presence within the local community. For building trust the trustor's propensity to trust and the trustee's perceived trustworthiness are of importance. An incremental Agile delivery strategy delivers software regularly, and prioritizes the most important features first (Agile-CSF6). Such a strategy allows for more user involvement (Law & Charron, 2005), which subsequently means that cooperation is an important prerequisite.

The remaining three critical success factors for Agile can be challenging to satisfy for similar reasons: team capability (Agile-CSF2), project management processes (Agile-CSF4), and Agile software engineering techniques (Agile-CSF5). As discussed in chapter 4, project management and ICT skills

are often lacking in developing countries (ICT4D-CSF3). If people from the community are actively included in the software development process, they will need to be educated on Agile. This relates to both Agile processes in project management and Agile software engineering techniques. Furthermore, because ICT4D is a multidisciplinary field (Thapa & Hatakka, 2017), not every team member may have a background in ICT. It is thus possible that people assigned to roles such as cultural interpreter or business strategy (to ensure economic self-sustainability) are not familiar with Agile either, and will also have to be educated. However, teaching Agile may not be difficult, or even necessary. According to the interviewees (see chapter 2.5, Appendix B), it is not Agile that needs to be taught, but rather the principles behind Agile. So for example, the concepts of collaboration and iterations. Furthermore, according to the interviewees, these principles are shared with fields of science relevant for ICT4D, for example social sciences. If that is the case, teaching Agile (or rather, the principles behind Agile) to team members without an ICT background may not be a problem. After the training is done, Agile project management processes and Agile software engineering techniques need to be applied. One possible difficulty that can arise here is the interference of donors. According to the interviewees, the commitment of the team is stronger to the donors (who are the customers) than to the local community (who are the users). In the experience of the interviewees, managers of donor companies prefer contracts, clear roadmaps, and traditional software development.

Team capability (Agile-CSF2) requires two further considerations to be made. First, the team members need to possess domain knowledge, or gain domain knowledge through interaction with the local community. The latter method will require cooperation. Second, in satisfying the critical success factors demand-driven (ICT4D-CSF2), local ownership (ICT4D-CSF5), and local partnerships (ICT4D-CSF7; chapter 4.2.3), motivation of the local community for the project is built. Both domain knowledge and motivation are important aspects of team capability.

In summary, several critical success factors for ICT4D projects have to be considered before Agile can be used in ICT4D projects. First, the ICT4D project needs to be demand driven (ICT4D-CSF1). Second, Agile practices need to be taught to the local community members involved as well as to team members without an ICT background (ICT4D-CSF3). Third, a cultural understanding must be developed (ICT4D-CSF12). According to the interviewees (see chapter 2.5; Appendix B), an ICT4D project should start out by looking at what the local community has, and by letting the local community explain what they do. Field research is important in this initial step, and the goal is to determine in what ways ICT could be used. In this initial step a cultural understanding can be developed, thus allowing Agile to be used once development initiates. Fourth, trust must be built (ICT4D-CSF13). Similarly to developing a cultural understanding, trust can be built in that initial step. However, trust cannot be fully built, because it is in iterative and dynamic process according to the interviewees and Mayer et al. (1995, p. 728). In chapter 6.2 will be explained how Agile can contribute to building trust. If a demand-driven approach does not build enough motivation within the local community, local ownership (ICT4D-CSF5) and building local partnerships (ICT4D-CSF7) also becomes a prerequisite for using Agile. Building local partnerships may also help building a cultural understanding (see chapter 4.2.1). According to the interviewees, local partnerships may also help building trust. Finally, two incompatibilities between Agile and ICT4D exist. Agile may be incompatible for ICT4D projects in regards to team size. Special attention will thus need to be paid to the negative coordination and colocation effects of the extra team members and roles. Furthermore, Agile may be incompatible for ICT4D projects because it is not always possible to work on location.

6.2. Effects of Agile software development on ICT4D projects

The effects of Agile on ICT4D projects will be determined by analysing which advantages or characteristics of Agile can influence which critical success factors for ICT4D projects. For each

critical success factor for ICT4D projects will be determined if any of the advantages of Agile can have a positive effect. In Table 9 a summary of these effects can be found. In the middle column the critical success factors for ICT4D projects can be found, and their identifiers can be found in the left column. In the right column the effect of Agile can be found.

Identifier	Critical success factor for ICT4D	Does Agile have an effect?
ICT4D-CSF1	Monitor and evaluate regularly	Yes (Positive)
ICT4D-CSF2	An ICT4D project must be demand-driven	Yes (Positive)
ICT4D-CSF3	Relevant skills must be built and trained	Uncertain (Both positive and negative)
ICT4D-CSF4	Efforts must be made to retain staff	Yes (Positive)
ICT4D-CSF5	Project ownership must be given to local parties	Yes (Positive)
ICT4D-CSF6	An ICT4D project must be economically self-sustainable	Yes (Positive)
ICT4D-CSF7	Local partnerships must be built to achieve synergies	Insignificant
ICT4D-CSF8	The creation of local content must be facilitated	Yes (Positive)
ICT4D-CSF9	The political context must be analysed and considered	Insignificant
ICT4D-CSF10	An ICT4D project must have a project champion	Insignificant
ICT4D-CSF11	The right technology must be chosen	Yes (Positive)
ICT4D-CSF12	Develop a cultural understanding of the local community	Yes (Positive)
ICT4D-CSF13	Build trust between the local community and outside parties	Yes (Positive)

Table 9 – Effect of Agile on the critical success factors for ICT4D projects

An ICT4D project must be regularly monitored and evaluated (ICT4D-CSF1). The improved control over projects (Agile-ADV9) suggests that Agile can contribute to this critical success factor for ICT4D. In particular, the good internal communication of teams can be of benefit here (Agile-ADV5). The frequent meetings can (and should) be used to discuss the current progress of the ICT4D project. Furthermore, the frequent delivery of project deliverables (Agile-CSF6) and the feedback from users (Agile-ADV2) can help with regular evaluation.

In order to be successful, an ICT4D project must be demand-driven (ICT4D-CSF2). Agile has several advantages that can contribute to ensuring an ITC4D project is demand-driven. First, user collaboration is one of the key pillars of Agile (see Introduction), and as a result Agile has good communication with the user (Agile-ADV2). Elicitation of requirements is done iteratively and frequently, ensuring that the ICT fits the demand of the users. Second, should requirements change, which according to Haikin (2013) occurs more than normally for ICT4D projects (see chapter 3.1), then Agile has the advantage of being robust to change (Agile-ADV1). According to the interviewees (see chapter 2.5; Appendix B), demos, prototypes, workshops and movies make the local community familiar with ICT and helps them understand how ICT could help them in their lives.

In chapter 6.1 was discussed that using Agile methods will result in skills pertaining to Agile having to be built and taught in addition to other ICT and management skills (ICT4D-CSF3), thereby making that ICT4D critical success factor more time consuming to achieve. However, Agile might also help building and training Agile related skills and other ICT skills, because Agile methods foster organizational learning (Agile-ADV10). In particular, pair programming might help building programming skills. The net effect of Agile upon building and training relevant skills is thus unknown.

In regards to retaining the staff (ICT4D-CSF4) Agile might help. Two benefits of Agile are the increased job satisfaction of employees (Agile-ADV6) and increased organizational learning (Agile-ADV10). Egan, Yang, and Bartlett (2004) found that job satisfaction is negatively correlated to turnover intention (p. 293-294), which is to say increasing job satisfaction will reduce the intention

to leave the firm (or the project). Furthermore, Egan et al. found that an organizational learning culture is strongly, positively correlated with job satisfaction. Finally, the correlation between learning culture and turnover intention was not significant. Egan at al. conclude that an organizational learning culture is still a good construct to increase employee retention, because of organizational learning culture is indirectly linked to turnover intention through job satisfaction. Ferguson and Ballantyne (2002) also believe that increasing job satisfaction is important in order to retain staff.

Local ownership (ICT4D-CSF5), as discussed, is not only about actually owning the ICT, but also about active participation of the local community. According to Wu (2011), an important antecedent for participation is the user's perceived support by the organization (p. 871). This can be achieved by, for example, listening to the problems of the users and solving these problems (Hirschman, 1970, p. 42). Agile can thus indirectly increase participation of the local community due to the positive effect of Agile for ensuring an ICT4D project is demand-driven. Furthermore, the good communication with users (Agile-ADV2) can help with facilitating participation.

In chapter 6.1, developing a cultural understanding (ICT4D-CSF11) was concluded to be a prerequisite for Agile to work successfully. However, it is also worth it to look at benefits Agile can bring to developing a cultural understanding, in case a cultural understanding needs to be improved. One way of developing a cultural understanding is by having a strong presence within the local community (Sandeep & Ravishankar, 2016, p.17-18), which Agile contributes to by focussing on user collaboration (see the Introduction) and by improving user communication (Agile-ADV2). Furthermore, the good, internal communication (Agile-ADV5) and the improved organizational learning (Agile-ADV10) of Agile can help disseminate the cultural understanding throughout the organization.

When the concept of trust was discussed in chapter 4.2.4, so too was the way trust is built. To summarize, there are two important concepts: the trustor's propensity to trust and the trustee's perceived trustworthiness (Mayer et al., 1995). Agile cannot influence the trustor's propensity to trust, because there is no reason to believe Agile can influence factors such as personality or culture. Agile also cannot initially influence the trustee's perceived trustworthiness, because there is no reason to believe Agile can influence the ability of the trustee (although skills and expertise, such as domain knowledge, are required to make Agile work successfully, Agile-CSF1), the benevolence of the trustee, or the integrity of the trustee. However, the perceived trustworthiness of the trustee is dynamic and affected by the results of trust-taking behaviour of the trustor. In the context of ICT4D projects (ICT4D-CSF13), trust-taking behaviour of the trustor can be considered as letting an organization into the local community or allowing them to develop ICT that will change their lives. The iterative nature of Agile (see the Introduction) and the frequent delivery of working software (Agile-CSF6) will allow for more frequent outcomes of trust-taking behaviour. And the increased user satisfaction when using Agile methods (Agile-ADV4) suggests that these outcomes will be more frequently positive.

Agile might have some small effects on the following critical success factors for ICT4D projects. By contributing to other critical success factors for ICT4D projects, Agile helps ensure the continued use of and therefore the demand for the ICT. Sustainable demand is one of the pillars of an economically self-sustainable ICT4D project (ICT4D-CSF6). Additionally, by focussing on working software (see the Introduction) and by frequently delivering software (Agile-CSF6), local content (ICT4D-CSF8) can be created earlier on in the ICT4D project. Finally, because Agile contributes to ensuring a demand-driven ICT4D project, an improved understanding of the local community's needs is gained. This will allow the right technology to be chosen with greater accuracy (ICT4D-CSF11).

Agile was not believed to have any significant impact on three critical success factors for ICT4D. The collaborative nature of Agile (see the Introduction) might improve the communication with and satisfaction of partners (ICT4D-CSF7) and project champions (ICT4D-CSF10), similar as to how it improves the communication with and satisfaction of users (Agile-ADV2, Agile-ADV4). However, it cannot help with seeking partners and building partnerships, or with finding project champions. Finally, Agile was not believed to be able to analyse and consider the political context (ICT4D-CSF9). However, the adaptive nature of Agile (see the Introduction) and the subsequent robustness to changing requirements (Agile-ADV1) allows an ICT4D project to respond to changes in the political context.

In summary, in terms of frequency, the most important advantages and characteristics of Agile can be identified. Improved communication with the user (Agile-ADV2) has a positive effect on four critical success factors for ICT4D, improved organizational learning (Agile-ADV10) on three, and good communication within the team on two. Furthermore, while not defined as an advantage, the focus on frequent delivery (Agile-CSF6) has a positive effect on three critical success factors.

7. Discussion

The value of this thesis lies in the insight it gives into how Agile can improve ICT4D projects and in what ways. This is important both for Agile frameworks (for example Doerflingen and Dearden, 2013; Hansson et al., 2009) and for frameworks using Agile elements (for example, Bon et al., 2016; Ferrario et al., 2014).

This thesis has several limitations or assumptions that must be addressed. First, a limitation regarding the ICT4D literature. In the interview that was held (see chapter 2.5; Appendix B), the interviewees raised a potential problem in ICT4D literature. On one end there are case studies. While interesting, the question is what their findings mean for ICT4D as a whole. On the other end there is desk research. Such research attempts to create policy for ICT4D as a whole, but lacks a link to real ICT4D projects. The interviewees' opinion suggests that the ICT4D literature used in this thesis is inadequate to provide a conclusive answer to the research question. Furthermore, neither the literature review for related work in Agile methods for ICT4D, the literature collected for determining the critical success factors for ICT4D projects, or the literature determining the critical success factors for, the advantages of, and the limitations of Agile were exhaustive. Additionally, an important assumption must be discussed. The issue pertaining the difference between the user and the customer in an ICT4D project was raised in the Introduction. An assumption in this thesis is that Agile advantages or characteristics such as customer satisfaction and customer collaboration can be rephrased to user satisfaction and user collaboration without any consequences. Moreover, the connections between Agile and ICT4D, while having a basis in literature, were made with the insight of the author of this thesis. The author's inexperience with both Agile and ICT4D may result in possible missed connections, both positive and negative. Finally, one result from the interview (see chapter 2.5; Appendix B) is that Agile shares principles with fields of research relevant for ICT4D. For example, user collaboration is also of importance in human computer interaction and social science. This suggests there may exist additional literature that could have been relevant for answering the research question.

8. Conclusions and further research

The relationship between Agile and ICT4D was explored largely due to the focus of Agile on user collaboration. Theory postulated that there are additional variables to consider in an ICT4D project, and as such an answer to the following research question was sought: “To what degree can Agile software development improve ICT4D projects?”. This thesis arrived at an answer to the research question by answering four sub questions, the answers of which will be summarized below in chapter .

8.1. Conclusions

In this subsection an answer will be given to all sub questions from chapter 2.1. Collectively, these answers provide an answer to the main research question.

SQ1: *What are the critical success factors for ICT4D projects?*

A total of thirteen critical success factors were identified, with varying presence in literature. In an ICT4D project progress must be regularly monitored and evaluated, a demand-driven development process must be applied, relevant skills must be built and trained, staff must be retained, project ownership must be given to local parties, economic self-sustainability must be achieved, local partnerships must be built, creation of local content must be facilitated, the political context must be analysed and considered, the ICT4D project must have a project champion, the right technology must be chosen, a cultural understanding of the local community must be developed, and trust between the local community and outside parties must be built.

SQ2: *What are the critical success factors, advantages, and limitations for Agile software development?*

A total of six critical success factors, ten advantages, and two limitations were found. The critical success factors for Agile are the team environment, team capability, user involvement, project management, Agile software engineering techniques, and delivery strategy. The advantages of Agile are an increased robustness to changing requirements, improved communication with the user, a higher quality of software, increased user satisfaction, improved internal communication, improved employee job satisfaction, a higher return on investment, an increase in successful projects, improved control over projects, and improved organizational learning. The limitations of Agile are that it does not work well for distributed development, or for developing large, complex systems.

SQ3: *Can Agile methods successfully work in an ICT4D project?*

Using the results from SQ1 and SQ2, an analysis was performed on whether Agile methods could work successfully in ICT4D projects. The result of this analysis was that four critical success factors for ICT4D need to be satisfied before an Agile method can work: the projects needs to be demand-driven, skills pertaining to Agile need to be taught to the stakeholders actively involved in the development, a cultural understanding must be developed, and trust must be built. Though not necessarily prerequisites, local ownership and building local partnerships can also play an important role in ensuring that Agile can work correctly, by increasing the motivation of the local community. Local partnerships can also contribute to developing a cultural understanding and building trust. An incompatibility between Agile and ICT4D however, exists. An ICT4D project requires more team roles, whereas Agile requires small team sizes. The advice for parties who seek to set up a demand-driven ICT4D project is to select an Agile method for ICT4D projects that allows for a substantial pre-

development phase, in which relevant skills can be taught and developed, a cultural understanding can be developed, and trust can be built. However, the parties must also consider that all those three critical success factors are iterative processes. Therefore, the Agile method must also allow for efforts to be made towards satisfying those three critical success factors in later stages of the project.

SQ4: *How can ICT4D projects benefit from an Agile approach?*

From the outcome of an analysis made using the results from SQ1 and SQ2, it can be concluded that Agile can positively contribute towards satisfying all but four critical success factors for ICT4D: monitor and evaluate regularly, ensure a demand-driven ICT4D project, retain staff, local ownership, economic self-sustainability, creation of local content, choose the right technology, develop a cultural understanding, and build trust. For three critical success the effect is insignificant: build local partnerships, understand the political context, and ensure a project champion. For the remaining critical success factor, building and training skills, there is both a positive and negative effect, thus resulting in an uncertain net effect. The most important advantages or characteristics of Agile, in terms of frequency, are the improved communication with the user, improved organizational learning, good communication within the team, and the focus on frequent delivery.

8.2. Further research

According to Ciaghi & Villafiorita (2011), ICT4D frameworks should implement the best practices of software engineering techniques (such as Agile) (p. 5). Two of the four frameworks identified in chapter 3.2 did so. This thesis examined how Agile can improve ICT4D projects, but further research is necessary to do the same for other software engineering techniques, as well as to determine how different software engineering techniques can be successfully combined. Additionally, more research is necessary on to what degree the critical success factors for ICT4D projects are important for which ICT4D projects. The reasons for this are that occurrences of critical success factors vary among articles that seek to give an overview on these critical success factors (see chapter 4.3), and among the discussed case studies (see chapter 4.2). Finally, research is necessary to determine whether Agile advantages or characteristics involving the customer for ICT projects can be correctly attributed to both the customer and the user in an ICT4D project, as was assumed in this thesis (see also the Introduction, the Discussion).

References

Alias, Z., Zawawi, E. M. A., Yusof, K., & Aris, N. M. (2014). Determining critical success factors of project management practice: A conceptual framework. *Procedia-Social and Behavioral Sciences*, *153*, 61-69. doi: 10.1016/j.sbspro.2014.10.041

Ambler, S. (2005). Quality in an agile world. *Software Quality Professional*, *7*(4), 34.

Bhatt, P., Ahmad, A. J., & Roomi, M. A. (2016). Social innovation with open source software: User engagement and development challenges in India. *Technovation, 52*, 28-39.

Blake, E. (2010, May). Software engineering in developing communities. In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering* (pp. 1-4). ACM.

Bon, A., & Akkermans, H. (2014). Rethinking technology, ICTs and development: Why it is time to consider ICT4D 3.0. 1-18. Retrieved from https://w4ra.org/wp-content/uploads/2015/01/ICT4D3.pdf

Bon, A., Akkermans, H., & Gordijn, J. (2016). Developing ICT Services in a Low-Resource Development Context. *Complex Systems Informatics and Modeling Quarterly*, 9, 84-109.

Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *IEEE software*, *25*(1), 60-67.

Ceschi, M., Sillitti, A., Succi, G., & De Panfilis, S. (2005). Project management in plan-based and agile companies. *IEEE software*, *22*(3), 21-27.

Chib, A., & Komathi, A. L. E. (2009, April). Extending the technology-community-management model to disaster recovery: assessing vulnerability in rural Asia. In *Information and Communication Technologies and Development (ICTD), 2009 International Conference on* (pp. 328-336). IEEE.

Chong, J. (2005, July). Social Behaviors on XP and non-XP teams: A Comparative Study. In *Agile Conference, 2005. Proceedings* (pp. 39-48). IEEE.

Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of systems and software*, *81*(6), 961-971.

Ciaghi, A., & Villafiorita, A. (2011, November). Crowdsourcing ICTD best practices. In *International Conference on e-Infrastructure and e-Services for Developing Countries* (pp. 167-176). Springer Berlin Heidelberg.

Ciaghi, A., Molini, P., Villafiorita, A., & Weldemariam, K. S. (2013, May). Maputo living lab summer school of ICTs: An experience report. In *IST-Africa Conference and Exhibition (IST-Africa), 2013* (pp. 1-8). IEEE.

Ciaghi, A., Villafiorita, A., & Dalvit, L. (2014). Understanding Best Practices for ICTD Projects: Towards a Maturity Model. *In ICTs for Inclusive Communities in Developing Societies. Proceedings of the 8th International Development Informatics Association Conference* (pp. 349-360).

Conradie, D.P., Morris, C., & Jacobs, S.J. (2003). Using information and communication technologies (ICTs) for deep rural development in South Africa. Communicatio. 29(1&2), 199–217. doi: 10.1080/02500160308538027

De Munster, I. L. (2005). The digital divide in Latin America: a case study. *Collection Building*, *24*(4), 133-136. doi:10.1108/01604950510629309

Dearden, A., Rizvi, H., & Gupta, S. (2010). Roles and responsibilities in agile ICT for development. In *Electronic Workshops in Computing* (pp. 1-12). British Informatics Society.

Doerflinger, J. and Dearden, A. (2013) Evolving a Software Development Methodology for Commercial ICTD Projects. Information Technologies & International Development, 9 (3), 43-60

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, *50*(9), 833-859

Egan, T. M., Yang, B., & Bartlett, K. R. (2004). The effects of organizational learning culture and job satisfaction on motivation to transfer learning and turnover intention. *Human resource development quarterly*, *15*(3), 279-301.

Ferguson, J., & Ballantyne, P. (2002, May). Sustaining ICT-enabled development: Practice makes perfect? Paper presented at the meeting of the International Institute for Communication and Development, The Hague, Netherlands

Ferrario, M. A., Simm, W., Newman, P., Forshaw, S., & Whittle, J. (2014, May). Software engineering for 'social good': integrating action research, participatory design, and agile development. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 520-523). ACM.

Fowler, M. (1999). *Refactoring: Improving the Design of Existing Code*. Boston, USA: Addison-Wesley Professional.

Garland, D., & Shaw, M. (1993). An Introduction to Software Architecture. In V. Ambriola, & G. Tortora (Eds.), *Advances in Software Engineering and Knowledge Engineering* (vol. II, pp. 1-39). River Edge, USA: World Scientific Publishing Company.

Glock, C. H. (2017). Decision support models for managing returnable transport items in supply chains: A systematic literature review. *International Journal of Production Economics*, *183(B)*, 561-569.

Green, B. N., Johnson, C. D., & Adams, A. (2006). Writing narrative literature reviews for peer-reviewed journals: secrets of the trade. *Journal of chiropractic medicine*, *5*(3), 101-117.

Haikin, M. (2013, March). Reflections on applying iterative and incremental software development methodologies (Agile, RAD etc.) to aid and development work in developing countries [Blarticle]. Retrieved from http://www.hiidunia.com/wp-content/uploads/downloads/2013/07/agile-blarticle-part-11.pdf

Hanna, N. K. (2010). *e-Transformation: Enabling New Development Strategies*. doi:10.1007/978-1-4419-1185-8_3

Hansson, H., Mozelius, P., Suhonen, J., Sutinen, E., Vesisenaho, M., & Wettegren, G. (2009). ICT4D with a Nordic flavor—A stepwise and multithreaded approach. In *IST-Africa 2009 conference proceedings. IIMC International Information Management Corporation* (pp. 1-9).

Haxby, A., & Lekhi, R. (2017, May). Building Capacity in Kenya's ICT Market Using Cross-Border Scrum Teams. In *International Conference on Social Implications of Computers in Developing Countries* (pp. 359-366). Springer, Cham.

Heeks, R. (2008a). ICT4D 2.0: The next phase of applying ICT for international development. Computer, 41(6), 26–33. doi: 10.1109/MC.2008.192

Heeks, R. (2008b, October 19). eGovernment Success/Failure: Definitions. Retrieved April 13, 2017, from http://www.egov4dev.org/success/sfdefinitions.shtml

Heimgärtner, R., Solanki, A., & Hollerit, B. (2014, June). Enhancing Usability Engineering in Rural Areas Using Agile Methods. In A. Marcus (Ed.), *International Conference of Design, User Experience, and Usability* (pp. 445-452). Springer International Publishing.

Highsmith, J. (2002). What is Agile Software Development? *CrossTalk - The Journal of Defense Software Engineering*, *15*(10), 4-9.

Hilkka, M. R., Tuure, T., & Rossi, M. (2005). Is extreme programming just old wine in new bottles: A comparison of two cases. *Journal of Database Management*, *16*(4), 41.

Hirschman, A. O. 1970. *Exit, voice and loyalty: Responses to decline in firms, organizations, and states*, Cambridge, MA: Harvard University Press.

Holz, H., & Maurer, F. (2002, August). Knowledge management support for distributed agile software processes. In *International Workshop on Learning Software Organizations* (pp. 60-80). Springer Berlin Heidelberg.

Ilieva, S., Ivanov, P., & Stefanova, E. (2004, August). Analyses of an agile methodology implementation. In *Euromicro Conference, 2004. Proceedings. 30th* (pp. 326-333). IEEE.

Independent Evaluation Group. (2011). *Capturing Technology for Development: An Evaluation of World Bank Group Activities in Information and Communication Technologies Volume 1*. Retrieved from https://openknowledge.worldbank.org/bitstream/handle/10986/2370/653750PUB0v10B0BLIC00ict0evaluation.pdf

International Development Research Centre (IDRC); Council for the Development of Social Science Research in Africa (CODESRIA). <used the executive summary, not sure by whom it was written>

Jacobs, S.J., & Herselman, M.E. (2005). An ICT-Hub model for rural communities. International Journal of Education and Development using ICT, 1(3), 57–93

James, T. (Ed.). (2004). Networking institutions of learning – SchoolNet: Information and communication technologies for development in Africa (Vol. 3). Ottawa, Ontario, Canada; Dakar: the International Development Research Centre and the Council for the Development of Social Science Research

Janzen, D., & Saiedian, H. (2005). Test-driven development concepts, taxonomy, and future direction. *Computer*, *38*(9), 43-50.

Karlström, D., & Runeson, P. (2005). Combining agile methods with stage-gate project management. *IEEE software*, *22*(3), 43-49.

Keniston, K., & Kumar, D. (2003). The four digital divides. Delhi: Sage.

Krauss, K. (2016). Demonstrating Critically Reflexive ICT4D Project Conduct and ICT Training in Rural South Africa. *GlobDev 2016. Paper 13.*

Law, A., & Charron, R. (2005, May). Effects of agile practices on social factors. In *ACM SIGSOFT Software Engineering Notes* (Vol. 30, No. 4, pp. 1-5). ACM.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., ... & Zelkowitz, M. (2002, August). Empirical findings in agile methods. In D. Wells, & L. Williams (Eds), *Conference on Extreme Programming and Agile Methods* (pp. 197-207). Springer Berlin Heidelberg.

Liu, C. (2016). Sustainability of rural informatization programs in developing countries: A case study of China' s Sichuan province. *Telecommunications Policy*, *40*(7), 714-724.

Mancl, D., Fraser, S., Opdyke, B., Hadar, E., & Hadar, I. (2009, October). Architecture in an agile world. In *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications* (pp. 719-720). ACM.

Mahanti, A. (2006). Challenges in enterprise adoption of agile methods-A survey. *CIT. Journal of computing and information technology*, *14*(3), 197-206.

Manifesto for Agile software development. (2001). Retrieved 12-3-2017, from http://agilemanifesto.org/

Mann, C., & Maurer, F. (2005, July). A case study on the impact of scrum on overtime and customer satisfaction. In *Agile Conference, 2005. Proceedings* (pp. 70-79). IEEE.

Mannaro, K., Melis, M., & Marchesi, M. (2004, June). Empirical analysis on the satisfaction of IT employees comparing XP practices with other software development methodologies. In *International Conference on Extreme Programming and Agile Processes in Software Engineering* (pp. 166-174). Springer Berlin Heidelberg.

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of management review*, *20*(3), 709-734.

Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, *82*(11), 1869-1890.

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & Prisma Group. (2009). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS med*, *6*(7), e1000097.

Monge, P. R., & Contractor, N. S. (2003). *Theories of communication networks*. New York, USA: Oxford University Press.

Mphahlele, M.E.K., & Maepa, M.E. (2003). Critical success factors in telecentre sustainability: A case study of six telecentres in the Limpopo Province. Communicatio, 29(1&2), 218–232. doi: 10.1080/02500160308538028

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72-78.

Paulk, M. C. (2002). Agile Methodologies and Process Discipline. *CrossTalk - The Journal of Defense Software Engineering*, *15*(10), 15-18.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, *13*(3), 303-337.

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*(3), 262-281.

Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006). Can distributed software development be agile?. *Communications of the ACM*, *49*(10), 41-46.

Renken, J., & Heeks, R. (2013, December). Conceptualising ICT4D project champions. In *Proceedings of the Sixth International Conference on Information and Communications Technologies and Development: Notes-Volume 2* (pp. 128-131). ACM.

Rico, D. F. (2008). What is the Return on Investment (ROI) of agile methods. Retrieved 12-3-2017, from http://ww.davidfrico.com/rico08a.pdf

Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software*, *17*(4), 26-32. doi: 10.1109/52.854065

Sandeep, M. S., & Ravishankar, M. N. (2016). Impact sourcing ventures and local communities: a frame alignment perspective. *Information Systems Journal*, *26*(2), 127-155.

Sengupta, B., Chandra, S., & Sinha, V. (2006, May). A research agenda for distributed software development. In *Proceedings of the 28th international conference on Software engineering* (pp. 731-740). ACM.

Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management*, *33*(5), 1040-1051.

Shiang, C. W., Halin, A. A., Lu, M., & CheeWhye, G. (2016). Long Lamai Community ICT4D E-Commerce System Modelling: An Agent Oriented Role-Based Approach. *The Electronic Journal of Information Systems in Developing Countries*, *75(5)*, 1-22.

Siddaway, A. (n.d.). What is a systematic literature review and how do I do one? [University educational document]. Retrieved from https://www.stir.ac.uk/media/schools/management/documents/centregradresearch/How%20to%20do%20a%20systematic%20literature%20review%20and%20meta-analysis.pdf

Sillitti, A., Ceschi, M., Russo, B., & Succi, G. (2005, September). Managing Uncertainty in Requirements: A Survey in Documentation-Driven and Agile Companies. In *Proceedings of the 11th IEEE International Software Metrics Symposium* (p. 17-27). IEEE Computer Society.

Teka, D., Dittrich, Y., & Kifle, M. (2016, May). Usability challenges in an Ethiopian software development organization. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering* (pp. 114-120). ACM.

Tessem, B. (2003, May). Experiences in learning XP practices: A qualitative study. In *International Conference on Extreme Programming and Agile Processes in Software Engineering* (pp. 131-137). Springer Berlin Heidelberg.

Thapa, D., & Hatakka, M. (2017). Introduction to "ICT4D: ICTs and Sustainable Development" Minitrack. *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2579.

Touray, A., Salminen, A., & Mursu, A. (2013). ICT barriers and critical success factors in developing countries. *The Electronic Journal of Information Systems in Developing Countries*, *56*(7), 1-17.

Turk, D., France, R., & Rumpe, B. (2014). Limitations of agile software processes. Third International Conference on Extreme Programming and Flexible Processes in Software Engineering (XP2002), May 26-30, Alghero, Italy, pp. 43-46, 2002.

United Nations Development Programme (UNDP) (2001). Essentials: Information communication technology for development (Evaluation Office No. 5). New York, NY: United Nations Development Programme.

United Nations Educational, Scientific, and Cultural Organization (UNESCO) (2011). UNESCO ICT Competency Framework for Teachers. Paris: United Nations Educational, Scientific and Cultural Organization (UNESCO).

Van Teijlingen, E. (2014). *Semi-structured interviews* [Powerpoint slides]. Retrieved from https://intranetsp.bournemouth.ac.uk/documentsrep/PGR%20Workshop%20-%20Interviews%20Dec%202014.pdf

W4RA. (n.d. -a). About W4RA. Retrieved from https://w4ra.org/w4ra/

W4RA. (n.d. -b). RadioMarché, Voice-based market information system. Retrieved from https://w4ra.org/radiomarche-voice-based-market-information-system/

Walsham, G. (2017). ICT4D research: reflections on history and future agenda. *Information Technology for Development*, *23*(1), 18-41. doi:10.1080/02681102.2016.1246406

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE software*, *17*(4), 19-25.

Wu, C. H. J. (2011). A re-examination of the antecedents and impact of customer participation in service. *The Service Industries Journal*, *31*(6), 863-876.

Appendix A – Literature review

In this appendix the process of the literature review during the four PRISMA Statement phases (as described in chapter 2.4) will be further explained. In Table 10 an example of the results from a literature review query can be found. In the first two rows information regarding the search engine and the query is given. The other ten rows are the first ten results from the query. If results are not retrievable then that means the articles are behind a paywall.

Item	Explanation	Retrievable?
Search engine	Google Scholar	
Query	"Agile software development" "ICT4D" framework	
Result #1	Ferrario, M. A., Simm, W., Newman, P., Forshaw, S., & Whittle, J. (2014, May). Software engineering for 'social good': integrating action research, participatory design, and agile development. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 520-523). ACM.	Yes
Result #2	Doerflinger, J., & Dearden, A. (2013). Evolving a software development methodology for commercial ICTD projects. *Information Technology and International Development*, *9*(3), 43-60.	Yes
Result #3	Alsmadi, I., & Saeed, S. (2013). A software development process for open source and open competition projects. *International Journal of Business Information Systems, 12*(1), 110-122.	Yes
Result #4	Bon, A., & Akkermans, H. (2014). Rethinking technology, ICTs and development: Why it is time to consider ICT4D 3.0. Retrieved from https://w4ra.org/wp-content/uploads/2015/01/ICT4D3.pdf	Yes
Result #5	Bhatt, P., Ahmad, A. J., & Roomi, M. A. (2016). Social innovation with open source software: User engagement and development challenges in India. *Technovation, 52*, 28-39.	Yes
Result #6	Dearden, A., & Haider Rizvi, S. M. (2015). ICT4D and participatory design. *The International Encyclopedia of Digital Communication and Society*.	No
Result #7	Pscheidt, M., & van der Weide, T. (2009). Supporting the ARIS community system in Mozambique. In *Prato Community CIRN Conference: Empowering Communities: Learning from Community Informatics Practice, Prato, Italy.*	Yes
Result #8	Ciaghi, A., Villafiorita, A., & Dalvit, L. (2014). Understanding Best Practices for ICTD Projects: Towards a Maturity Model. *In ICTs for Inclusive Communities in Developing Societies. Proceedings of the 8th International Development Informatics Association Conference* (pp. 349-360).	Yes
Result #9	Steyn, J., & Van Greunen, D. (2015). *ICTs for inclusive communities in developing societies*. Newcastle upon Tyne, The United Kingdom: Cambridge Scholars Publishing.	No
Result #10	Bon, A., Gordijn, J., & Akkermans, H. (2016). A low-resource aware framework for ICT service development in rural Africa. *Proceedings of the CAiSE'16 Forum, at the 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016), 1612*, 177-184.	Yes

Table 10 – Literature review: example of a search query

In addition to the queries, the following article was suggested by one of the study researchers after contacting him (dr. Akkermans): Bon, Akkermans, & Gordijn (2016).

In total, 47 articles were identified using the method described in chapter 2.4. This already excludes duplicates. Furthermore, the search engine IEEE returned a few table of contents, which were pre-emptively excluded. All queries were performed on the 28th of May 2017.

During the screening phase, more articles were removed. First, the articles behind a paywall were removed because these could not be accessed. Then, an assessment was made regarding the relevancy of each articles based on the title and the abstract, as well as an automatic search for the terms "Agile" and "ICT4D". For example, the title of result #3 from Table 10: "A software development process for open source and open competition projects", made it clear that the article was about open source and open competition projects, and not ICT4D projects. An automatic search on the term 'ICT4D' confirmed this: the only returned result was about how ICT4D was an interest of the study's author. The article was thus removed during the screening process.

Determining the eligibility of the articles remaining after the screening phase went according to the eligibility criteria as specified in chapter 2.4. Not all articles that did not pass the eligibility phase were discarded. Some articles were able to serve other purposes, for example Ciaghi & Villafiorita (2011) and Dearden, Rizvi, & Gupta (2010). However, such articles are not included in the list of eligible articles.

The eligible articles together form the included articles. In Table 11 the full list of included articles can be found. In the left column the items are numbered, and in the right column the reference is given. These articles are discussed in chapter 3.

Item	Reference
Article #1	Bhatt, P., Ahmad, A. J., & Roomi, M. A. (2016). Social innovation with open source software: User engagement and development challenges in India. *Technovation, 52*, 28-39.
Article #2	Bon, A., & Akkermans, H. (2014). Rethinking technology, ICTs and development: Why it is time to consider ICT4D 3.0. 1-18. Retrieved from https://w4ra.org/wp-content/uploads/2015/01/ICT4D3.pdf
Article #3	Bon, A., Akkermans, H., & Gordijn, J. (2016). Developing ICT Services in a Low-Resource Development Context. *Complex Systems Informatics and Modeling Quarterly*, 9, 84-109.
Article #4	Ciaghi, A., Molini, P., Villafiorita, A., & Weldemariam, K. S. (2013, May). Maputo living lab summer school of ICTs: An experience report. In *IST-Africa Conference and Exhibition (IST-Africa), 2013* (pp. 1-8). IEEE.
Article #5	Ciaghi, A., Villafiorita, A., & Dalvit, L. (2014). Understanding Best Practices for ICTD Projects: Towards a Maturity Model. *In ICTs for Inclusive Communities in Developing Societies. Proceedings of the 8th International Development Informatics Association Conference* (pp. 349-360).
Article #6	Doerflinger, J., & Dearden, A. (2013). Evolving a software development methodology for commercial ICTD projects. *Information Technology and International Development*, *9*(3), 43-60.
Article #7	Ferrario, M. A., Simm, W., Newman, P., Forshaw, S., & Whittle, J. (2014, May). Software engineering for 'social good': integrating action research, participatory design, and agile development. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 520-523). ACM.
Article #8	Haikin, M. (2013, March). Reflections on applying iterative and incremental software development methodologies (Agile, RAD etc.) to aid and development work in

	developing countries [Blarticle]. Retrieved from http://www.hiidunia.com/wp-content/uploads/downloads/2013/07/agile-blarticle-part-11.pdf
Article #9	Hansson, H., Mozelius, P., Suhonen, J., Sutinen, E., Vesisenaho, M., & Wettegren, G. (2009). ICT4D with a Nordic flavor—A stepwise and multithreaded approach. In *IST-Africa 2009 conference proceedings. IIMC International Information Management Corporation* (pp. 1-9).
Article #10	Haxby, A., & Lekhi, R. (2017, May). Building Capacity in Kenya's ICT Market Using Cross-Border Scrum Teams. In *International Conference on Social Implications of Computers in Developing Countries* (pp. 359-366). Springer, Cham.
Article #11	Heimgärtner, R., Solanki, A., & Hollerit, B. (2014, June). Enhancing Usability Engineering in Rural Areas Using Agile Methods. In A. Marcus (Ed.), *International Conference of Design, User Experience, and Usability* (pp. 445-452). Springer International Publishing.
Article #12	Teka, D., Dittrich, Y., & Kifle, M. (2016, May). Usability challenges in an Ethiopian software development organization. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering* (pp. 114-120). ACM.

Table 11 – Literature review: list of articles included

Appendix B – Interview topics

In this appendix the topics of the interview will be given, as well as an overview of where in the thesis the information of the interview is used. The interview began with the interviewer and the interviewees getting acquainted with each another. Permission was asked to record the interview (which was given), the interviewer explained the purpose of the interview, and the interviewees introduced themselves. In Table 12 the predetermined topics can be found, and in Table 13 the topics that were also discussed but not planned can be found. For both tables, the topics can be found in the left column and the location in the thesis in the right column. Two of the predetermined topics were not covered in the interview, because a suitable moment to ask about these topics was not encountered. The conversation about organizational learning moved into the direction of team communication quickly, resulting in no significant amount of information regarding organization learning.

Topic	Location in the thesis
Teaching Agile skills to team members and the local community	Chapter 6.1
Developing a cultural understanding	Chapter 6.1
Building trust	Chapter 6.1
Benefits of user participation	Not covered in interview
Organizational learning	Not discussed in thesis
Team communication	Chapter 6.1
Frequent delivery of software	Not covered in interview

Table 12 – Predetermined topics for the interview

Topic	Location in the thesis
The difference between the user and the customer	Introduction
The commitment to the donor	Chapter 6.1
Shared principles between Agile and fields of science relevant for ICT4D	Chapter 6.1, Discussion
The first phase of an ICT4D project	Chapter 6.1
Prototyping	Chapter 6.2
Criticism of ICT4D literature	Discussion

Table 13 – Additional topics for the interview that were not planned