Improving ICT4D projects with Agile software development

Joost Dijkers
Utrecht University
Faculty of Science, Department of Information and Computing Sciences
Utrecht, the Netherlands
J.J.G.Dijkers@students.uu.nl

Sietse Overbeek
Utrecht University
Faculty of Science, Department of Information and Computing Sciences
Utrecht, the Netherlands
S.J.Overbeek@uu.nl

Sergio España
Utrecht University
Faculty of Science, Department of Information and Computing Sciences
Utrecht, the Netherlands
S.Espana@uu.nl

ABSTRACT

ICT4D seeks to bridge the digital divide in developing countries. Important requirements of ICT4D projects are a demand-driven approach and participation of the local community. The fact that user collaboration is a principle of Agile software development (Agile), triggers our interest on whether Agile practices can improve ICT4D projects. This paper aims to investigate if and how Agile can contribute to the success of ICT4D projects. In order to achieve this, existing literature was consulted and an interview was held. This paper provides an overview of the critical success factors for ICT4D projects and Agile, as well as of the advantages of Agile. Agile can only work successfully when ICT4D projects are demand-driven, and when both a cultural understanding and trust are built. Notable 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, ICT4D, digital divide, user collaboration

ACM Reference Format:
Joost Dijkers, Sietse Overbeek, and Sergio España. 2018. Improving ICT4D projects with Agile software development. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

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 for short [1]. ICT4D is aimed at how the benefits of ICT can be evenly divided between society to bridge the gap between the rich and poor. For example, ICT can improve creating, sharing, and enhancing knowledge, make production and transactions more efficient and cost-effective, and stimulate networking amongst parties (e.g. firms) [2]. However, high rates of failure exist for ICT4D projects [3, 4].

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Agile Software Development (henceforth referred to as Agile) is a methodology for developing software and was found to increase the success rate of ICT projects [5]. Agile is collaborative, incremental, and iterative [6]. Collaborative development 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 development approach in which the system is developed in a series of small steps. Iterative development means that the development activities, such as requirements engineering and software testing, are performed cyclically rather than sequentially. Furthermore, Agile is adaptive, which means that rapid change is supported [7]. Agile practices are summarized by Highsmith [7] as follows: 'short iterations, continuous testing, selforganizing teams, constant collaboration (…), and frequent replanning based on current reality'.

However, agile methods that have harvested success in western countries cannot be directly applied in ICT4D projects [8]. There exist multiple reasons for this. For example, increased user participation has proven to be essential in order to achieve ICT adoption [9]. Additionally, inhabitants of poor communities in developing countries often have no ICT or project management skills [10]. Furthermore, cultural barriers can limit or even prevent the cooperation of the local community [11, 12]. Research is thus necessary on to what degree Agile methods are compatible with ICT4D projects. In addition, research on the effects of using Agile methods in ICT4D projects is limited, as it is mainly focused on benefits of user collaboration. For example, using an Agile method was reported to allow developers to change the system in a natural way in response to unexpressed requirements and changes in business environment [13]. Furthermore, Agile methods make ICT more demand-driven and improve the involvement of users [14, 15]. This increased user participation of Agile methods also allows requirements to be elicited and knowledge to be created [14, 16]. To address these two problems, the following research question is formulated:

RQ: *To what degree can Agile software development improve ICT4D projects?*

The research question is answered by first performing a literature study on the critical success factors for ICT4D projects, the critical success factors for Agile methods, and the advantages of Agile methods. Then, the results of the literate study are analyzed.

The structure of this paper is as follows. In section 2 related work on Agile methods for ICT4D projects is discussed to find out why Agile is used in frameworks for ICT4D projects. In section 3 the method for arriving at an answer to the research question is given. In section 4 the results from the literature study on the critical success factors for ICT4D projects and Agile methods, and

on the advantages of Agile methods are presented. In section 5 these results are analyzed to determine how suitable Agile methods are for ICT4D projects and in what ways they can improve ICT4D projects. In section 6 the analysis is discussed and related to existing literature. Finally, in section 7 an answer is given to the research question and suggestions for future research are given.

2 RELATED WORK

Bon, Akkermans, and Gordijn developed an ICT4D framework that is partially based on Agile [18]. Other inspirations of the framework are: Living Labs, use case analysis, and requirements engineering. The framework is specifically designed to address several ICT4D concerns, such as a lack of understanding of the local needs and the context. The discussed benefits of Agile are that it fosters creativity, personal commitment, and collaboration with the user.

Distributed Agile Methodology Addressing Technical Ictd in Commercial Settings (DRAMATICS) is an Agile method for commercial ICT4D projects [19]. The discussed benefit of Agile is the collaboration with users.

Speedplay is a framework for ICT4D projects which takes inspiration from Agile, Action Research, and Participatory Design [20]. Some of the inspirations from Agile are iterative development, flexibility, and collaborative development. The discussed benefit of Agile is the user collaboration.

The Nordic Model is a framework for ICT4D based on Nordic socio-cultural background and shared values, and is described as an Agile method [21]. 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.

These frameworks differ in terms of other inspirations (e.g. Participatory Design for Speedplay) or application (e.g. business ICT projects for DRAMATICS). However, all these frameworks have been tested in ICT4D projects with success. The success of Agile or Agile inspired frameworks suggests that Agile methods can be beneficial to ICT4D projects [22]. The primary reason for using Agile methods for ICT4D seems to be improved collaboration with the user.

3 METHOD

3.1 Research sub-questions

This paper aims to answer the main research question: 'To what degree can Agile software development improve ICT4D projects?'. It does so by answering the following research sub-questions (SQ):

SQ1: *Can Agile methods successfully work in an ICT4D project?*

An analysis is performed to determine if Agile can work successfully in ICT4D projects. It is essential for Agile to work successfully in order for the ICT4D project to succeed [24]. In order to analyze this, critical success factors for ICT4D projects and Agile are gathered. Critical success factors (CSF) are defined by Alias, Zawawi, Yusof, and Aris [23] as: 'Inputs to project management practice which can lead directly or indirectly to project success'. As such, CSFs give a good impression of where Agile methods might have an important effect.

SQ2: *How can ICT4D projects benefit from an Agile approach?*

The benefits Agile methods can bring specifically to ICT4D projects are discussed in order to understand how Agile can improve ICT4D projects. Data regarding the advantages of using Agile is gathered to answer this research sub-question.

3.2 Data gathering

These sub questions are answered by using existing literature. Google Scholar was primarily used as search engine, as well as IEEE Computer Society Digital Library and SpringerLink. Some common search terms that were used are: 'Agile software development', 'Agile software development advantages', 'Agile software development critical success factors', 'ICT4D', 'ICT4D critical success factor', and 'ICT4D Agile'. The forward snowballing technique was used as well.

CSFs for ICT4D projects were in literature referred to with the following terms: critical success factor, lesson learned, step (to ensure sustainable development), and activity (that led to success). For Agile, the following two terms were found and used to describe CSFs: critical success factor, lesson learned. Advantages of using Agile were sometimes also called benefits.

3.3 Data analysis

To answer the first research sub-question, the CSFs for Agile are discussed in the context of ICT4D projects and by relating these to the CSFs for ICT4D projects, resulting in a series of steps that need to be taken in setting up an ICT4D project before Agile can be applied. To answer the second research sub-question, a comparison between the advantages of Agile and the CSFs for ICT4D projects was made. For each CSF for ICT4D projects it was determined if Agile can improve the degree to which that CSF is satisfied.

3.4 Interview

In order to attain a greater insight into how Agile can improve ICT4D projects, an interview with ICT4D and Agile experts was held. A semi-structured interview is a good choice when the purpose of the interview is to elicit a person's viewpoint regarding a specific matter [25]. In a semi-structured interview, there are predetermined questions, but there is flexibility in asking these questions. For example, new questions can be added ad hoc. The results from the interview were used to validate the findings from the literature study, and are thus discussed in the Analysis section.

The interviewees are all part of the organization W4RA, of which the name stands for the Web alliance for Regreening in Africa. On its website, W4RA gives its mission as follows [26]: '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 [27], 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'.

4 RESULTS

4.1 Critical success factors for ICT4D projects

Monitor and evaluate project progress regularly (ICT4D-CSF1) [10, 28, 29]. Monitoring and evaluating project progress allows for the team members to measure the effects of ICT on development [29]. 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. Monitoring and evaluating also allows for problems to be identified earlier, which, if acted upon, can ensure a more effective and efficient project [10].

An ICT4D project must be demand-driven (ICT4D-CSF2) [28–32]. The ICT4D project must satisfy the present needs while also allowing for the needs of future generations [31]. Furthermore, implementing an ICT4D project in an area where there is not sufficient demand will not result in a sustainable ICT4D project [28]. A critical element in achieving this 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.

Relevant skills must be built and trained (ICT4D-CSF3) [10, 28–31, 33]. Project management, implementation, and ICT skills are scarce in developing countries and need to be taught [10]. Illiteracy is also an important problem [34]. Building and training these skills can be expensive however, so mechanisms for knowledge sharing to reduce costs are recommended [31]. ICT training also helps overcome technophobia [35]. Finally, this training be a continuous process [28].

Efforts must be made to retain staff (ICT4D-CSF4) [10, 31]. The effects of talented staff leaving can be disastrous [31]. Other than traditional intrinsic (e.g. praise) and extrinsic (e.g. salary) rewards, project ownership (ICT4D-CSF5) can be a major motivational incentive due to the involvement it brings.

Project ownership must be given to local parties (ICT4D-CSF5) [28, 29, 31]. Local ownership is defined as the active participation of the local community in all phases of the development process [31]. Successful local ownership will result in the community viewing the ICT as an integral part of their daily lives [28]. Local ownership is related to two others CSFs: 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).

An ICT4D project must be economically self-sustainable (ICT4D-CSF6) [10, 12, 28–32]. Many ICT4D projects rely on donor money for their continued survival, which means these projects risk falling apart as soon as enthusiasm and funding from outside partners disappears [32]. Economic self-sustainability is therefore important to ensure the long-term success of ICT4D projects. However, donor money is important initially, because due to the experimental nature of many ICT4D projects it cannot be expected for these projects to be profitable from the get-go [18, 32]. One particular important aspect to ensure economic self-sustainability is marketing, because the inability to inform the community about the benefits of ICT4D projects is one of the main reasons why ICT4D projects fail [28]

Local partnerships must be built to achieve synergies (ICT4D-CSF7) [10, 28, 29, 31]. 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 [29, 31].

The creation of local content must be facilitated (ICT4D-CSF8) [10, 29, 30, 32, 33]. Local content is content being in local language as well as having inspiration from local culture, created by locals [32]. 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. An example of what local content can be is information for farmers regarding which vegetables can be grown on their fields [30].

The political context must be analyzed and considered (ICT4D-CSF9) [10, 29, 31–33]. The political situation in a country can affect an ICT4D project on two levels: micro and macro level [31]. 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.

An ICT4D project must have a project champion (ICT4D-CSF10) [10, 28, 29, 36]. Renken and Heeks [36] define an ICT4D project champion 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'. Multiple ICT4D project champions are necessary, to reduce the risk of the project falling apart if an ICT4D project champion leaves the project [10].

The right technology must be chosen (ICT4D-CSF11) [10, 29–31, 33]. 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 key factors [31]. Technology also needs to be affordable for the people involved with the ICT4D project [10].

A cultural understanding of the local community must be developed (ICT4D-CSF12) [11, 33]. Cultural understanding can be necessary to avoid conflicts during the constant interaction between outsiders and the local community [11]. Cultural understanding can also be necessary to become accepted within the local community and gain their trust, as well as to gain access to their resources [12].

Trust between the local community and outside parties must be built (ICT4D-CSF13) [11, 12]. 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 [37]. In an ICT4D project the local community would be the trustor, and the outside party the trustee. There are two factors that determine the level of trust [37]: the trustor's propensity to trust and the trustee's perceived trustworthiness. The propensity to trust differs among individuals, but factors that influence the propensity are history with development, personality, and the culture. Trustworthiness has ability, benevolence, and integrity as antecedents. Ability refers to the skills and expertise of a party within a domain (e.g. knowledge about ICT). Benevolence refers to what degree the party desires to help the trustor without regard to

extrinsic rewards. Finally, integrity refers the trustee's adherence to principles.

4.2 Critical success factors for Agile software development

A survey study to the CSFs of Agile projects on four dimensions to project success revealed six CSFs for Agile methods [24]. 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). The six CSFs are discussed below.

Team environment (Agile-CSF1). A good team environment contributes positively to the quality of the product [24]. 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 [38, 39].

Team capability (Agile-CSF2). Team capability positively contributes to the timeliness and cost of a project [24]. 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. Additionally, relevant technical training should be provided to the team members. Highly competent team members are important to compensate for the smaller team size [38]. Finally, developers must possess domain knowledge in order to be able to communicate with the users [40].

User involvement (Agile-CSF3). User involvement positively contributes to the scope of the product [24]. 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.

Project management (Agile-CSF4). Project management processes positively contributes to the quality of the product [24]. 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.

Agile software engineering techniques (Agile-CSF5). Agile software engineering techniques positively contribute to the quality and scope of the product [24]. These techniques are: coding standards, simple design, refactoring, limited but sufficient documentation, and integration testing.

Delivery strategy (Agile-CSF6). Delivery strategy positively contributes to the scope of the product, and the timeliness and cost of the project [24]. A good delivery strategy prioritizes the important features of the product first. Furthermore, software should be regularly delivered.

4.3 Advantages of Agile software development

More robust to changing requirements (Agile-ADV1). Requirements are inherently variable because both the developer and user acquire more knowledge about the domain of the application [41]. Requirements also change because the business environment in which the user is positioned is dynamic [42]. Agile methods are more robust to change than traditional methods because of two reasons [41]. 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.

Improved communication with the user (Agile-ADV2). In Agile methods, face-to-face communication with the user instead of rigorous documentation is the norm [42]. The iterative nature of Agile allows for more frequent communication with the user. Furthermore, this more frequent face-to-face-communication allows for an improved elicitation and validation of requirements, which reduces the likelihood of requirements changing later on [42, 43].

Higher quality of software (Agile-ADV3). Software quality principles that quality professionals have been preaching for are included in Agile methods [6]. 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 [44]. Frequent user feedback is also mentioned as a reason for improved software quality [39].

Increased user satisfaction (Agile-ADV4). There are several factors that contribute to this [45]: improved communication with users (Agile-ADV2), increased user involvement (Agile-CSF3), and the improved quality of software (Agile-ADV3).

Good, internal communication (Agile-ADV5). The required strong focus on internal communication (Agile-CSF4) results in an improved understanding of the requirements, tasks, project status, and resource allocation among all team members [43].

Improved employee job satisfaction (Agile-ADV6). Job satisfaction is higher for Agile methods for six reasons [46]: 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. The improved software quality (Agile-ADV3) also contributed to a higher job satisfaction [45].

A higher return on investment (Agile-ADV7). The return on investment (ROI) is higher in projects done with Agile methods for several reasons [47]: higher software quality (Agile-ADV3), increased user satisfaction (Agile-ADV4), lower costs, and higher productivity.

Increase in successful projects (Agile-ADV8). The more of the principles of the Agile approach is applied in the project, the higher the project success [5]. Hayes' study (as cited in [39]) attributed this occurrence to the iterative nature of Agile. An iterative cycle instead of a sequential cycle supposedly increases the 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.

Improved control over projects (Agile-ADV9). Hayes (as cited by Mahanti [39]) argues that Agile methods improve the

control over projects due to several reasons: 'Short iterations, multidisciplinary teams, knowledge sharing, continuous integration, and feedback'.

Improved organizational learning (Agile-ADV10). Agile methods focuses on teamwork and foster organizational learning within those teams [48], for example with pair programming [49].

5 ANALYSIS

5.1 Suitability of Agile software development for ICT4D projects

The suitability of Agile for ICT4D projects is analyzed by comparing the CSFs for Agile methods and for ICT4D projects.

A good team environment (Agile-CSF1) is also necessary in order for an ICT4D project to be successful. For ICT4D projects it is beneficial for all the team members to be located near each other, 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). However, this is not always possible for ICT4D projects. For example, according to the interviewees, 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, because it is not financially doable to remain there during the entire project. Another aspect of the team environment is that teams should be small, which fits with ICT4D projects because of budget restrictions.

User involvement (Agile-CSF3) requires cooperation with the local community, which is also important for ICT4D projects (ICT4D-CSF2). From the 'Critical success factors for ICT4D projects' section, 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 [51], which subsequently means that cooperation is an important prerequisite.

The following three CSFs 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). 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 [50], 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, 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 CSFs demand-driven (ICT4D-CSF2), local ownership (ICT4D-CSF5), and local partnerships (ICT4D-CSF7), motivation of the local community for the project is built. Both domain knowledge and motivation are important aspects of team capability.

In summary, several CSFs 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, 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 [37], which is also echoed by the interviewees. 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 (ICT4D-CSF7). According to the interviewees, local partnerships may also help building trust. Finally, one incompatibility between Agile and ICT4D exists. Agile may be incompatible for ICT4D projects because it is not always possible to work on location.

5.2 How Agile software development can improve ICT4D projects

The effects of Agile on ICT4D projects are determined by analyzing which advantages or characteristics of Agile can influence which CSFs for ICT4D projects.

Monitor and evaluate project progress regularly (ICT4D-CSF1). The improved control over projects (Agile-ADV9) suggests

that Agile can contribute. 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.

An ICT4D project must be demand-driver (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, 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 occurs more than normally for ICT4D projects [17], then Agile has the advantage of being robust to change (Agile-ADV1). According to the interviewees, demos, prototypes, workshops and movies make the local community familiar with ICT and helps them understand how ICT could help them in their lives.

Relevant skills must be built and trained (ICT4D-CSF3). One result from section 5.1 is that skills pertaining to Agile have to be built and taught as well, thus making this CSF 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). For example, pair programming helps build programming skills. The net effect of Agile upon building and training relevant skills is thus unknown.

Efforts must be made to retain staff (ICT4D-CSF4). Two benefits of Agile that can help are the increased job satisfaction of employees (Agile-ADV6) and increased organizational learning (Agile-ADV10). One research found that job satisfaction is negatively correlated to turnover intention, which is to say that increasing job satisfaction will reduce the intention to leave the firm (or the project) [52]. Furthermore, that same research 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. However, organizational learning culture is still a good construct to increase employee retention, because organizational learning culture is indirectly linked to turnover intention through job satisfaction [52].

Project ownership must be given to local parties (ICT4D-CSF5). An important antecedent for user participation is the user's perceived support by the organization [53]. This can be achieved by, for example, listening to the problems of the users and solving these problems [54]. Agile can 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.

A cultural understanding of the local community must be developed (ICT4D-CSF12). A strong presence within the local community helps develop a cultural understanding [12], which Agile contributes to by focusing on user collaboration 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.

Trust between the local community and outside parties must be built (ICT4D-CSF13). To summarize how trust is built, there are two important concepts: the trustor's propensity to trust and the trustee's perceived trustworthiness [37]. 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, 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 impact their lives. The iterative nature of Agile 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 CSFs for ICT4D projects. By contributing to other CSFs 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 focusing on working software 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 CSFs 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.

Additionally, 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 CSFs 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 CSFs.

6 DISCUSSION

ICT4D projects have high rates of failure [3, 4] and the Agile methodology was found to increase the success rate of ICT projects [5]. However, Agile methods successful in western countries cannot carelessly be applied in ICT projects for developing countries [8]. Examples of why are the worse or lack of ICT and management

skills [10] and cultural barriers [11, 12]. Furthermore, current research on why to apply Agile methods is mainly limited to the benefit of improved user collaboration [14–16].

The main value of this research lies in the insight it gives into how Agile can improve ICT4D projects and in what ways. This is important both for Agile frameworks as (for example [19, 21]) and for frameworks using Agile elements (for example [18, 20]), because there might be additional aspects of Agile that could further improve the frameworks discussed in the related works section or form the theoretical basis of new ICT4D frameworks. Existing literature mainly focuses on user collaboration as a reason for why Agile methods can improve ICT4D projects [14–16], whereas this paper found other aspects of Agile as well that can improve ICT4D projects. Most notable are organizational learning, team communication, and frequent delivery.

However, several limitations to this research must be addressed. First, a limitation regarding the ICT4D literature. In the interview that was held, 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 thus suggests that the ICT4D literature used in this paper is inadequate to provide a conclusive answer to the research question. Furthermore, literature for the CSFs for ICT4D and Agile and for the advantages of Agile was not collected in a systematic way, thus providing no guarantee that the CSFs of ICT4D and Agile and the advantages of Agile are exhaustive. A final limitation is that the findings of this research have not been validated in practice or through rigorous expert interviews.

7 CONCLUSION

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 paper arrived at an answer to the research question by answering two sub-questions, the answers of which are summarized below. Collectively, these answers provide an answer to the main research question.

SQ1: *Can Agile methods successfully work in an ICT4D project?*

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. 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.

SQ2: *How can ICT4D projects benefit from an Agile approach?*

Agile can positively contribute towards satisfying all but four critical success factors for ICT4D: monitor and evaluate the project regularly, ensure a demand-driven ICT4D project, make efforts to retain staff, give local owernship to the local community, ensure economic self-sustainability, create 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.

As addressed in the discussion, the results of this research have not been validated in practice. Case studies are thus necessary in which Agile methods are used, so that the proposed benefits can be assessed. Furthermore, as ICT4D is a multi-disciplinary field, it might prove worthwhile to investigate the effects of other software engineering techniques or approaches from other sciences, which can then be used to design an ICT4D framework.

8 ACKNOWLEDGEMENTS

The authors wish to thank prof. dr. Akkermans, ms. drs. Bon, and ms. drs. Tuijp for sharing their expertise during the interview.

REFERENCES

[1] Geoff Walsham. 2017. ICT4D research: reflections on history and future agenda. *Information Technology for Development* 23, 1 (February 2017), 18–41. DOI:http://dx.doi.org/10.1080/02681102.2016.1246406

[2] Nagy K. Hanna. 2010. *E-Transformation: Enabling New Development Strategies*, New York, NY: Springer-Verlag New York.

[3] Richard Heeks. 2002. Information Systems and Developing Countries: Failure, Success, and Local Improvisations. *The Information Society* 18, 2 (2002), 101–112. DOI:http://dx.doi.org/10.1080/01972240290075039

[4] Independent Evaluation Group. 2011. Capturing Technology for Development: An Evaluation of World Bank Group Activities in Information and Communication Technologies Volume 1. (2011). Retrieved March 6, 2018 from https://openknowledge.worldbank.org/bitstream/handle/10986/2370/653750PUB0v10B0BLIC00ict0evaluation.pdf

[5] Pedro Serrador and Jeffrey K. Pinto. 2015. Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management* 33, 5 (2015), 1040–1051. DOI:http://dx.doi.org/10.1016/j.ijproman.2015.01.006

[6] Scott Ambler. 2005. Quality in an agile world. *Software Quality Professional* 7, 4 (2005), 34-40

[7] Jim Highsmith. 2002. What is Agile Software Development? *Crosstalk - The Journal of Defense Software Engineering* 15, 10 (2002), 4-9

[8] Joerg doerflinger and Tom Gross. 2010. Bottom billion architecture. Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development - ICTD 10 (2010). DOI:http://dx.doi.org/10.1145/2369220.2369228

[9] Joerg Doerflinger and Tom Gross. 2010. Technical ICTD - A User Centered Lifecycle. Communications: Wireless in Developing Countries and Networks of the Future IFIP Advances in Information and Communication Technology (2010), 72–83. DOI:http://dx.doi.org/10.1007/978-3-642-15476-8_8

[10] Tina James. 2004. Information and communication technologies for development in Africa, Ottawa: International Development Research Centre (IDRC).

[11] Kirstin Krauss. 2016. Demonstrating Critically Reflexive ICT4D Project Conduct and ICT Training in Rural South Africa. Proceedings Annual Workshop of the

AIS Special Interest Group for ICT in Global Development at AIS Electronic Library, paper 13.
[12] M.S. Sandeep and M.N. Ravishankar. 2015. Impact sourcing ventures and local communities: a frame alignment perspective. Information Systems Journal 26, 2 (2015), 127–155. DOI:http://dx.doi.org/10.1111/isj.12057
[13] Punita Bhatt, Ali J. Ahmad, and Muhammad Azam Roomi. 2016. Social innovation with open source software: User engagement and development challenges in India. Technovation 52-53 (2016), 28–39. DOI:http://dx.doi.org/10.1016/j.technovation.2016.01.004
[14] Rüdiger Heimgärtner, Alkesh Solanki, and Bernd Hollerit. 2014. Enhancing Usability Engineering in Rural Areas Using Agile Methods. Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience Lecture Notes in Computer Science (2014), 445–452. DOI:http://dx.doi.org/10.1007/978-3-319-07668-3_43
[15] Degif Teka, Yvonne Dittrich, and Mesfin Kifle. 2016. Usability challenges in an Ethiopian software development organization. Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 16 (2016). DOI:http://dx.doi.org/10.1145/2897586.2897604
[16] Anna Bon and Hans Akkermans. 2014. Rethinking technology, ICTs and development: Why it is time to consider ICT4D 3.0. (2014). Retrieved March 8, 2018 from https://w4ra.org/wp-content/uploads/2015/01/ICT4D3.pdf
[17] Matt Haikin. 2013. Reflections on applying iterative and incremental software development methodologies (Agile, RAD etc.) to aid and development work in developing countries. Retrieved March 8, 2018 from http://www.hiidunia.com/wp-content/uploads/downloads/2013/07/agile-blarticle-part-11.pdf
[18] Anna Bon, Hans Akkermans, and Jaap Gordijn. 2016. Developing ICT Services in a Low-Resource Development Context. Complex Systems Informatics and Modeling Quarterly, 9 (2016), 84–109. DOI:http://dx.doi.org/10.7250/csimq.2016-9.05
[19] Joerg Doerflinger and Andy Dearden. 2013. Evolving a Software Development Methodology for Commercial ICTD Projects. Information Technologies & International Development 9, 3 (2013), 43-60.
[20] Maria Angela Ferrario, Will Simm, Peter Newman, Stephen Forshaw, and Jon Whittle. 2014. Software engineering for social good: integrating action research, participatory design, and agile development. Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014 (2014). DOI:http://dx.doi.org/10.1145/2591062.2591121
[21] Henrik Hansson, Peter Mozelius, Jarkko Suhonen, Erkki Sutinen, Mikko Vesisenaho, and Gunnar Wettegren. 2009. ICT4D with a Nordic flavor - A stepwise and multithreaded approach. In IST-Africa 2009 conference proceedings. IIMC International Information Management Corporation, 1-9.
[22] Aaron Ciaghi, Adolfo Villafiorita, and Lorenzo Dalvit. 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, 349-360
[23] Zarina Alias, E.m.a. Zawawi, Khalid Yusof, and N.m. Aris. 2014. Determining Critical Success Factors of Project Management Practice: A Conceptual Framework. Procedia - Social and Behavioral Sciences 153 (2014), 61–69. DOI:http://dx.doi.org/10.1016/j.sbspro.2014.10.041
[24] Tsun Chow and Dac-Buu Cao. 2008. A survey study of critical success factors in agile software projects. Journal of Systems and Software 81, 6 (2008), 961–971. DOI:http://dx.doi.org/10.1016/j.jss.2007.08.020
[25] Edwin Van Teijlingen. 2014. Semi-structured interviews. (2014). https://intranetsp.bournemouth.ac.uk/documentsrep/PGR%20Workshop%20-%20Interviews%20Dec%202014.pdf
[26] W4RA. About W4RA. Retrieved March 11, 2018 from https://w4ra.org/w4ra/
[27] W4RA. RadioMarché, Voice-based market information system. Retrieved March 11, 2018 from https://w4ra.org/radiomarche-voice-based-market-information-system/
[28] M.E.Kutu Mphahlele and Maisela E. Maepa. 2003. Critical success factors in telecentre sustainability: a case study of six telecentres in the Limpopo Province. Communicatio 29, 1-2 (2003), 218–232. DOI:http://dx.doi.org/10.1080/02500160308538028
[29] United Nations Development Programme (UNDP). 2001. Essentials: Information communication technology for development. (2001). http://web.undp.org/evaluation/documents/essentials_5.pdf
[30] David Pieter Conradie, C. Morris, and S.J. Jacobs. 2003. Using information and communication technologies (ICTs) for deep rural development in South Africa. Communicatio 29, 1-2 (2003), 199–217. DOI:http://dx.doi.org/10.1080/02500160308538027
[31] Julie Ferguson, Peter Ballantyne, and Galin Kora. 2002. Sustaining ICT-enabled development practice makes perfect? In Sustaining ICT-enabled development practice makes perfect? The Hague: International institute for communication and development (IICD).
[32] Kenneth Keniston and Deepak Kumar. 2003. The Four Digital Divides, Delhi: SAGE Publishers.
[33] Almamy Touray, Airi Salminen, and Anja Mursu. 2013. ICT Barriers and Critical Success Factors in Developing Countries. The Electronic Journal of Information Systems in Developing Countries 56, 1 (2013), 1–17. DOI:http://dx.doi.org/10.1002/j.1681-4835.2013.tb00401.x
[34] Aaron Ciaghi and Adolfo Villafiorita. 2011.Crowdsourcing ICTD best practices. In International Conference on e-Infrastructure and e-Services for Developing Countries, 167-176
[35] Arul Chib and A.l.e Komathi. 2009. Extending the Technology-Community-Management model to disaster recovery: Assessing vulnerability in rural Asia. 2009 International Conference on Information and Communication Technologies and Development (ICTD) (2009). DOI:http://dx.doi.org/10.1109/ictd.2009.5426694
[36] Jaco Renken and Richard Heeks. 2013. Conceptualising ICT4D project champions. Proceedings of the Sixth International Conference on Information and Communications Technologies and Development Notes - ICTD 13 - volume 2 (2013). DOI:http://dx.doi.org/10.1145/2517899.2517928
[37] Roger C. Mayer, James H. Davis, and F. David Schoorman. 1995. An integrative model of organizational trust. Academy of management review 20, 3 (1995), 709-734.
[38] Mikael Lindvall et al. 2002. Empirical Findings in Agile Methods. Extreme Programming and Agile Methods — XP/Agile Universe 2002 Lecture Notes in Computer Science (2002), 197–207. DOI:http://dx.doi.org/10.1007/3-540-45672-4_19
[39] Aniket Mahanti. 2006. Challenges in Enterprise Adoption of Agile Methods - A Survey. Journal of Computing and Information Technology 14, 3 (2006), 197. DOI:http://dx.doi.org/10.2498/cit.2006.03.03
[40] Hilkka Merisalo-Rantanen, Tuure Tuunanen, and Matti Rossi. 2005. Is Extreme Programming Just Old Wine in New Bottles. Journal of Database Management 16, 4 (2005), 41–61. DOI:http://dx.doi.org/10.4018/jdm.2005100103
[41] Alberto Sillitti, Martina Ceschi, Barbara Russo, an Giancarlo Succi. 2005. Managing Uncertainty in Requirements: A Survey in Documentation-Driven and Agile Companies. 11th IEEE International Software Metrics Symposium (METRICS05). DOI:http://dx.doi.org/10.1109/metrics.2005.29
[42] Lan Cao and Balasubramaniam Ramesh. 2008. Agile requirements engineering practices: An empirical study. IEEE software 25, 1 (2008), 60-67
[43] Minna Pikkarainen, Jukka Haikara, Outi Salo, Pekka Abrahamsson, and Jari Still. 2008. The impact of agile practices on communication in software development. Empirical Software Engineering 13, 3 (2008), 303–337. DOI:http://dx.doi.org/10.1007/s10664-008-9065-9
[44] David Janzen and Hossein Saiedian. 2005. Test-driven development concepts, taxonomy, and future direction. Computer 38, 9 (2005), 43–50. DOI:http://dx.doi.org/10.1109/mc.2005.314
[45] Chris Mann and Frank Maurer. A case study on the impact of scrum on overtime and customer satisfaction. Agile Development Conference (ADC05). DOI:http://dx.doi.org/10.1109/adc.2005.1
[46] Katiuscia Mannaro, Marco Melis, and Michele Marchesi. 2004. Empirical Analysis on the Satisfaction of IT Employees Comparing XP Practices with Other Software Development Methodologies. Extreme Programming and Agile Processes in Software Engineering Lecture Notes in Computer Science (2004), 166–174. DOI:http://dx.doi.org/10.1007/978-3-540-24853-8_19
[47] David F. Rico. 2008. What is the Return on Investment (ROI) of Agile Methods? . (2008). http://ww.davidfrico.com/rico08a.pdf
[48] Sridhar Nerur, Radhakanta Mahapatra, and George Mangalaraj. 2005. Challenges of migrating to agile methodologies. Communications of the ACM 48, 5 (January 2005), 72–78. DOI:http://dx.doi.org/10.1145/1060710.1060712
[49] Harald Holz and Frank Maurer. 2003. Knowledge Management Support for Distributed Agile Software Processes. Advances in Learning Software Organizations Lecture Notes in Computer Science (2003), 60–80. DOI:http://dx.doi.org/10.1007/978-3-540-40052-3_7
[50] Devinder Thapa and Mathias Hatakka. 2017. Introduction to ICT4D: ICTs and Sustainable Development Minitrack. In Proceedings of the 50th Hawaii International Conference on System Sciences, 2579
[51] Amy Law and Raylene Charron. 2005. Effects of agile practices on social factors. Proceedings of the 2005 workshop on Human and social factors of software engineering - HSSE 05 (2005). DOI:http://dx.doi.org/10.1145/1083106.1083115
[52] Toby Marshall Egan, Baiyin Yang, and Kenneth R. Bartlett. 2004. The effects of organizational learning culture and job satisfaction on motivation to transfer learning and turnover intention. Human Resource Development Quarterly 15, 3 (2004), 279–301. DOI:http://dx.doi.org/10.1002/hrdq.1104
[53] Cedric Hsi-Jui Wu. 2011. A re-examination of the antecedents and impact of customer participation in service. The Service Industries Journal 31, 6 (2011), 863–876. DOI:http://dx.doi.org/10.1080/02642060902960768
[54] Albert O. Hirschman. 1970. Exit, voice and loyalty: Responses to decline in firms, organizations, and states. Cambridge, MA, Harvard University Press