Evolving a software development methodology for commercial ICTD projects

DOERFLINGER, Joerg and DEARDEN, Andy <http://orcid.org/0000-0002-5706-5978>

Research Article

Evolving a Software Development Methodology for Commercial ICTD Projects

Joerg Doerflinger
Andy Dearden

Joerg Doerflinger
joerg.doerflinger@sap.com
PhD Student
SAP, Mobile Research
Vincenz-Priessnitz-Strasse
Karlsruhe, BW
Germany
+49 6227 752556

Andy Dearden
a.m.dearden@shu.ac.uk
Professor of Interactive
Systems Design
Communication & Computing
Research Centre
Sheffield Hallam University
Cantor Building
153 Arundel Street
Sheffield, S1 2NU
UK
+44 114 225 6878

Abstract

This article discusses the evolution of a "DistRibuted Agile Methodology Addressing Technical Ictd in Commercial Settings" (DRAMATICS) that was developed in a global software corporation to support ICTD projects from initial team setup through ICT system design, development, and prototyping, to scaling up and transitioning, to sustainable commercial models. We developed the methodology using an iterative Action Research approach in a series of commercial ICTD projects over a period of more than six years. Our learning is reflected in distinctive methodology features that support the development of contextually adapted ICT systems, collaboration with local partners, involvement of end users in design, and the transition from research prototypes to scalable, long-term solutions. We offer DRAMATICS as an approach that others can appropriate and adapt to their particular project contexts. We report on the methodology evolution and provide evidence of its effectiveness in the projects where it has been used.

Introduction

Technical Information and Communication Technology for Development (ICTD) research refers to work by computer scientists and engineers to create appropriate ICT solutions to support social and economic development objectives. Critical challenges are not only to build contextually adapted ICT solutions, but also to find approaches for developing and deploying innovative systems in ways that are scalable and sustainable in the longer term (Brewer et al., 2005).

While the ICTD literature offers a number of successful examples of technical concepts adapted to the specific infrastructural and cultural context (Anokwa, Ribeka, Parikh, Borriello, & Were, 2012; Hartung et al., 2010; Kumar, 2004; Liu & Payne, 2011; Parikh & Lazowska, 2006; Veeraraghavan, Yasodhar, & Toyama, 2007; Zainudeen, Samarajiva, & Sivapragasam, 2011), literature on targeted software development methodologies is relatively sparse. Many researchers have reported that established (western) methods often do not fit the specific ICTD context "out of the box" (Heeks, 2009; Winschiers, 2006). However, some authors do report on ways to adapt existing methods to ICTD contexts (Blake & Tucker, 2006; Dearden & Rizvi, 2009; Dearden, Rizvi, & Gupta, 2010; Maunder, Marsden, Gruijters, & Blake, 2007; Sharpey-Schafer, 2009).

New innovation models have also been suggested to support long-term development goals, taking into account economic, infrastructure, environmental, and cultural contexts (Blake, 2010; Heeks, 2008; Thinyane & Terzoli, 2009).

In this article we describe a "DistRibuted Agile Methodology Addressing Technical Ictd in Commercial Settings" (DRAMATICS) that we have developed in several ICTD projects in the retail and agricultural supply chain sectors in Africa. The method informs our practice from initial research prototyping to sustainable and scalable long-term deployments. Using an iterative, action-research approach, we have adapted and refined the methodology through several projects. Lessons learned over more than six years are reflected in distinctive methodology features, such as the management of setup operations, the distribution and interaction of roles, and the different iteration structures used in different project phases.

In this article, we offer details of DRAMATICS as an example of a repeatable ICTD software development methodology that others may learn from, adapt, and appropriate to fit their particular contexts. DRAMATICS has been developed in projects that share a common theme of commercial supply chain management, but we believe that its principles may be adaptable to meet the demands of ICTD projects in other domains.

In the rest of the article, we describe the scope of our research interventions, present the Action Research approach used for iterative refinement, and discuss how DRAMATICS has evolved to its present state. We evaluate DRAMATICS by its instantiation and outputs in four ongoing pilot scenarios.

1. Background

DRAMATICS was developed in the context of a global software company whose goals include building long-term business relationships with partners in developing regions and opening new markets for the corporation. Collaborating in ICTD research projects that are partially funded by external organizations (e.g., European Commission, Bill & Melinda Gates Foundation), we establish collaborations with local partners to design, develop, and test ICT solutions that are appropriate to the context and that offer a potential return on investment for the corporation in the long term. During the research project we demonstrate solutions in large-scale field trials, explore long-term collaborations with local partners, and plan the transition to sustainable business. It is in the interests of both the research demonstration and the business to codify a systematic approach to software development that can be re-used across demonstration sites and by commercial divisions of the business in the future. Over more than six years we have refined DRAMATICS within several pilot implementations in three ICTD research projects: rural retail in South Africa (Collaboration@Rural—C@R), cashew traceability in Ghana and Burkina Faso (African Cashew initiative—ACi), and shea traceability (Star Shea Network—SSN) in Burkina Faso.

DRAMATICS was first trialed in C@R (Merz, De Louw, & Ullrich, 2007), aiming to foster socio-economic development of small, medium, and microenterprises (SMMEs). In South Africa there are about 130,000 so-called Spaza Shops (Ligthelm, 2002)—small retail enterprises selling basic goods to the local community. They face three major challenges: (1) Due to their remoteness, buying stock involves high transportation costs; (2) a procurement trip can take about one business day, closing the shop and incurring significant opportunity costs for other livelihood activities; and (3) because of limited liquidity, they can only buy small quantities in short buying cycles. Using our initial version of DRAMATICS, we developed an ICT system to support ordering goods by using mobile phones.

With ACi starting in mid-2009, DRAMATICS was refined and implemented in pilots in Ghana and Burkina Faso. ACi aims to increase the competitiveness of African cashew production in five West African countries (see Figure 1). In cooperation with local partners, business processes were redesigned and new ICT systems developed (Doerflinger & Gross, 2012) to provide traceability and transparency of the flow of cashews along the value chain, enabling organizations to obtain Fair Trade certification and associated premiums.

In late 2011, the concepts from the cashew pilot were re-used for a shea nut/butter traceability pilot in collaboration with the SSN—a national federation of shea farmer cooperatives in northeast Ghana. The participating shea collectors, organized in groups of 20–30, collect shea nuts and sell them at buying stations where the nuts are stored for transportation

Figure 1. Initial paper-based data management setup.

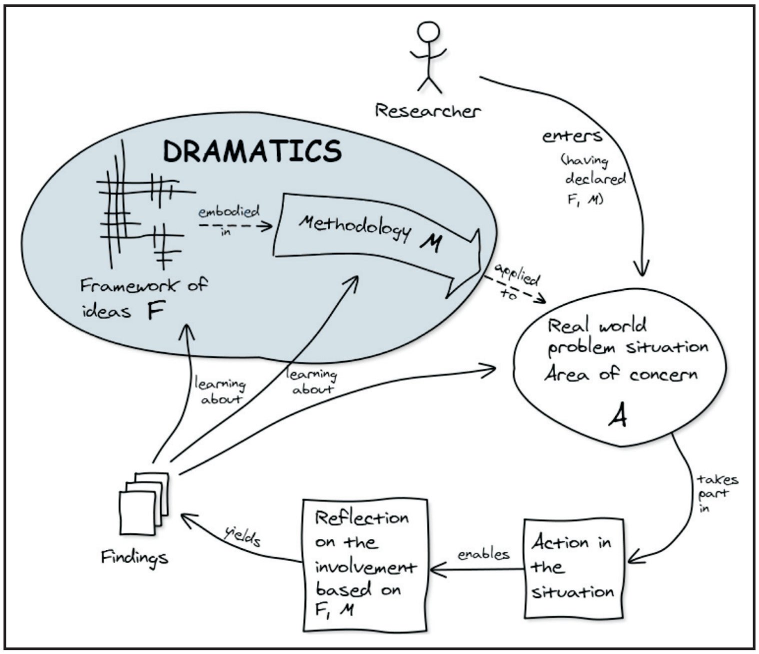

Figure 2. Checkland and Holwell's FMA approach illustrated for the case of DRAMATICS (based on Figures 2 and 3 in Checkland & Holwell, 1998).

to processing plants or buyers.[1] The shea business has process and traceability inefficiencies similar to the cashew business.

The continuous refinement of DRAMATICS across these projects was based on an iterative Action Research process which is described below.

2. Formative Refinement Approach

Action Research (AR) is an established framework for information systems research (Baskerville & Myers, 2004), and is gaining acceptance in computing research that seeks social relevance (Hayes, 2011). Building on Checkland and Holwell's (1998) approach in establishing the Soft Systems Methodology (SSM), we followed a cyclic Action Research (AR) approach to iteratively refine DRAMATICS. West and Stansfield (2001) describe this as the "FMA approach" (see Figure 2). In this approach, research proceeds as a series of cycles where researchers select an area of concern (A), the researchers then take action in that situation, applying the methodology (M), which is guided by a framework of ideas (F). It is important that F and M are made explicit prior to taking action. During and after each cycle, reflection generates findings that then lead to adaptations to the framework of ideas and, consequently, to the Methodology.

Our *Reflection* approach can be related to the NIMSAD (Normative Information Model-based Systems Analysis and Design) framework (Jayaratna, 1994). NIMSAD's aim is to evaluate the *Methodology* on three levels: the Problem Situation (Context), the Problem Solver (Methodology User), and the Problem-Solving Process (Methodology). In each *Reflection* phase we used questions covering these three topic areas. Questions such as "Was end-user feedback biased by cultural influences, e.g., trying to follow the strongest voice in a workshop?" explore the appropriateness of techniques in the *Problem Situation (Context).* Issues related to the *Problem Solver (Methodology User)* are explored through questions like "Did users approach you also with embarrassing questions?" or "Did users also criticize?" Questions like: "Did the outcomes/findings from this particular activity justify the cost?" or "Did the methodology ensure high quality of

1. *Shea trees grow wild in northern Ghana. During the rainy season, women (predominantly) journey into the bush to collect the nuts for sale to supplement family incomes.*

results during the field trial?" help to evaluate the *Problem Solving Process* (the Methodology itself).

Reflection happens in a variety of forms, including: collaborative face-to-face or online workshops between the change manager, supporting interns, and partner organizations/local contacts; team meetings of software developers, change manager, and interns; reflections by individual researchers, e.g., the change manager after end-user workshops. During reflection sessions findings are noted and then enriched with contextual descriptions and on-site images for discussion in subsequent meetings. Reflection workshops with partner organizations and local contacts take place after the prototype development phase, after installation and training in the field, and during productive use (e.g., after each harvesting season).

This reflective approach to reviewing the methods by the project team together with local partners aims to meet the standards of "recoverability" described by Checkland and Holwell (1998), as well as "dependability" and "confirmability" demanded by Hayes (2011). The ultimate goal of AR is the "workability" (Hayes, 2011) of an ICTD software development methodology from the software organization and local partners' perspectives. By *workability* in this context, we are concerned with finding a way of working such that systems can be developed that are sustainable and scalable in the context, and can be delivered within cost and time limits that are (commercially) realistic for the core partners.

It is important to acknowledge that AR is less central in this work than in some uses of it for technologies in social settings (e.g., Blake & Tucker, 2006; Byrne, 2003; Hayes, 2011). Hayes (2011) characterizes action researchers as "friendly outsiders" facilitating a community developing and executing research activities for itself around issues that the community selects for itself. Hayes' examples involve an academic acting as the AR facilitator, in contrast to the commercial setting of DRAMATICS. For Blake and Tucker (2006), each software iteration is an AR cycle, with mutual learning to strengthen the community's understanding of technology and shared understandings of community goals. Refining their encounter methodology is then a second-order action learning cycle. In our approach, reflective AR is used by the core project team and local partners to refine the methodology, but the individual software development iterations in DRAMATICS are not explicitly AR-based. Rather, they follow a variety of participatory techniques led by a core team.

3. Evolution of DRAMATICS

DRAMATICS has evolved over six years from our initial assumptions toward a refined, reused, and evaluated methodology. In the following text, we explain the initial methodology (M) and the framework of ideas that underpin it (F) before discussing its evolution and lessons learned through use.

3.1 Initial Methodology and Framework

Informed by ICTD literature and reports of previous ICTD projects, we expected to benefit from User-Centered and Participatory Design approaches. Thus, we started with an iterative development model using established interaction design methods such as rapid/iterative prototyping, on-site field studies, personas and scenarios, end-user workshops, interviews, questionnaires, and in-situ direct observation.

We expected that methods would need to be adapted to the infrastructural and cultural context (Winschiers, 2006). We recognized that we needed a tight collaboration with local stakeholders from the beginning (Brewer et al., 2005), and we knew that some form of collaboration contracts and guiding documentation would be needed. We also assumed that being on site physically to build relationships with stakeholders would be important. Thus, in addition to typical team roles like a project manager and software developers, we created a specific role for a locally based researcher. We also assumed that field visits by key team members would improve understanding for the whole team.

We expected that these ideas would be refined through experience, and below we discuss some of the key lessons that shaped the development of DRAMATICS as we present it in this article.

3.2 Collaboration Setup

In the initial framework we recognized that formal collaboration agreements would be needed with partners in our trials. In C@R we used adapted versions of the corporation's standard Test and Evaluation Agreement, which covers various aspects of nondisclosure and fair treatment. However, as C@R grew and eventually brought in a second supplier, issues of commercial confidentiality between the two suppliers had to be worked into the contracts.

The ACi involved more stakeholders, more complex relations between stakeholders, and direct provision of some hardware (smartphones at buying stations). Collaboration agreements had to specify the type and scale of collaboration (e.g., duration, number of systems deployed, usage rights of developed systems, hardware and field costs), what the corporation provides (e.g., manuals, system hosting, data confidentiality, operational/technical support), and the responsibilities for field activities (e.g., parallel fallback processes, participation in feedback sessions). Some partner organizations also had more mature legal departments, so the contracts also covered standard issues such as warranties and arrangements for contract termination or dispute. We have now developed three template collaboration contracts for different levels of involvement: (1) A Tester trials ICT without active participation in co-innovation; (2) a Co-Innovator trials and actively participates in design, with the expectation that the ICT will be customized for its needs and possible future discounts; (3) an Early Customer collaborates in the transition from research to real business. Our first Early Customer contract is expected in approximately two years when our solutions have matured.

In addition to the template contracts, we have developed reusable resources such as presentations to support recruitment and a data checklist for mapping partner organizations, selected processes, infrastructure, and resources.

Digitization of paper-based information (which is typical) is crucial in all our systems. In C@R, the on-site research team collected participant data (21 Spaza Shops and one retailer). In the ACi, with thousands of farmers, this approach was impractical. Instead, we prepared a printed template to collect farmer data (e.g., name, location, membership ID, and a photo) and a spreadsheet template to transcribe the data. We benefited from the (unpaid) support of local partners (e.g., buying agents who were visiting farmers anyway) and volunteers (e.g., international aid organization members) to conduct data capture and initial data entry. Internal software developers conducted data cleaning and integration.

3.3 Team Roles

Our initial assumptions created a role for a locally based research team member, but the relationship between this role and other team members was undefined. In C@R we had different people present on site, for example, if the locally based researcher was unavailable, the project manager or a software developer would travel to execute on-site experiments. Conversely, if extra software development was needed, the locally based researcher would spend time programming. We found that having different people on site resulted in weak trust relationships with local partners, added extra "hand-over" effort on return, and different individual interpretations of results.

During C@R, based on acquired skills, experiences, and personal preferences, team members gradually specialized to separate the work of the locally based researcher, whom we now describe as the "change manager," from the software development roles. For example, in C@R one of the *infopreneurs* (van Rensburg, 2008), a local contact offering basic IT services to the rural community, realized that some of the delivery drivers were becoming uncooperative. The infopreneur explained this to the locally based researcher, enabling the researcher to explore the concerns, discovering that some drivers feared job loss. Just one workshop with the drivers resolved the issue. Having a single contact on site regularly and continuously helps to establish friendly personal relationships. Local partners know the responsible contact person, and both parties develop skills in interpreting nonverbal cues and subtleties in conversations. By the start of ACi, we had clearly defined assigned roles and responsibilities.

Our framework of ideas thus evolved to highlight the importance of who did what. Like others (e.g., Byrne, 2003), we found that when local partners and local volunteers executed on-site questionnaires and interviews, end users were more open and honest than with foreign researchers. For example, in an early cashew pilot, the change manager visited buying station agents to gather usability feedback. The agents reported that they preferred the system because it improved efficiency. However, transaction reports in the next weeks showed that most agents were not using the system when cashew sacks from the warehouse were loaded onto trucks. Only when local contacts (aid agency representatives) approached them were we able to discover that the user interface (UI) at that time was actually hampering the loading process.

Our initial framework recognized ourselves as outsiders in a new context, carrying implicit, unconscious assumptions. For example, in C@R our early paper prototypes used maps and geographical information systems, and we were surprised to discover that what we considered "natural" actually comes from a cultural experience not shared by our end users.

While we recognized that we needed to be sensitive and continuously question our assumptions, we did not know how to execute this in practice. From the start we made this questioning part of the evaluation in each cycle. Due to this continuous sensitivity training, we became more reflexive in our day-to-day action, and recognized that personal reflexivity is a key skill for all the research roles. For example, in ACi's early UI design phase, a requirement collected in the field was to display transactions of all warehouses of a cooperative on the phone. However, during a UI mock-up review meeting between software developers and the change manager, we reflected on the designs and changed the UI to only display information to the warehouse manager relating to his individual warehouse. While we initially assumed that it would be beneficial to display all available information, we recognized during reflection that it could result in envy among warehouse agents and hamper the process. This was confirmed by end users when we deployed the first prototypes in the field.

3.4 Communication Flows

An important corollary of the team role specialization is that communications within the team and between the team and the local setting are now more structured. In C@R, besides the communication between the change manager and local partners, we also had direct communication between software developers and local partners, for example, discussing software issues directly via phone or online conferences. This produced redundancy and confusion about design specifications and was confusing for local partners. In our refined methodology, the change manager is the main contact for all local communication. Internal software developers communicate with local partners via the change manager.

Best results have been achieved when all the internal software developers were located in the same office. The larger scale and complexity of the ACi and SSN projects demanded that some work be outsourced to external software developers (other employees of the corporation based in other offices worldwide). External developers communicate solely with the internal developers, who extract and specify subtasks that do not require detailed contextual knowledge. External developers are used primarily in the initial prototype development and in bug-fixing.

3.5 Executing methods

Iterative reflections on methods have taught us much about local adaptation. For example:

- As a commercial concern, our framework of ideas must consider cost/benefit issues in all activities. In C@R, we executed a direct observation with three researchers and one student, including laptop and camera equipment, travelling roughly 500 kilometers in a rental car for a two-day field visit. The end users required transport to the evaluation venue, and catering was organized. In videos of these events, end users appeared to be uncomfortable, and the team worried that workshop results might be unreliable. For the same level of resources, local partners and contacts can execute several on-site activities in a fashion more comfortable for end users, providing greater overall value. Like Winschiers (2006) we found that end-user workshops, interviews, and questionnaires supported by local partners or local community members provided more reliable feedback than if led by the change manager alone.
- In our projects, a user-trains-user workshop format appears to make users more comfortable and improves effectiveness (and efficiency) when compared to a one-on-one trainer-trains-user setup. For example, in the ACi and SSN projects, we were able to reduce the number of trainers from up to 10 local supporters in early efforts to 2–3 supporters in current workshops.
- Because of unreliable power, paper prototypes and flip charts are important fallbacks to functional prototypes, but an early shift to functional prototypes improved feedback. For example, early in the ACi project, the change manager held a workshop for buying agents in a remote area. Since electricity was unreliable that day, he started the workshop using paper

prototypes. However, only with the shift to hands-on experiences with functional prototypes (for a very short time before batteries expired), end users provided far more feedback and asked far more questions.

- End users often try to follow the strongest voice or do not want to express doubts in front of community leaders or employers, so working with user groups of similar status is helpful.
- Finally, on-site activities must adhere to local cultural conventions such as praying, welcoming the community chief before a workshop starts, and respecting gender issues.

3.6 Iterative Process

Our initial methodology was an undifferentiated iterative process that included rapid prototyping and iterative refinement based on end-user feedback. This has evolved into a defined, three-phase approach to cope with the different needs of prototyping, roll-out, and live operation.

When the first robust prototypes are ready, we begin extensive on-site change management activities (e.g., end-user training workshops, hardware preparation and distribution, and implementing the incentive model). In this phase the ICT systems are extensively used by end-users in the real environment, thus combining beta testing and training in one phase. With initially no process established to handle the feedback, this resulted in end user confusion and difficulties in prioritizing bugs to fix and features to implement. During this phase we now implement a daily bug-fixing cycle based on reports from the change manager and local contacts. A parallel software development cycle implements minor changes and additional feature requests arising from live testing. This phase ends with the deployment of the ICT system for live use in the real business case.

In the productive phase, the iteration cycle adapts again. This phase requires operational support and maintenance (e.g., ensuring server availability, coping with hardware problems, assisting end users to recharge SIM cards), with software changes more gradual, that is, 1–2 months per cycle.

In the beginning we had one deployment platform for each of the three phases, resulting in mixed test and live data. We now operate three platforms with individual deployments for each pilot: a development platform, a testing platform, and a live platform.

In the next section, we present the DRAMATICS components in the form that we now use them (Figure 3).

4. Formative Refinement Result: DRAMATICS

Our presentation of the current version of DRAMATICS is split into two parts. Sections 4.1 and 4.2 cover the preconditions for DRAMATICS to be used and setup operations to prepare for DRAMATICS. Section 4.3 describes the process flow, collaboration arrangements, and communication flow. Section 4.4 describes the three-phase cyclic methodology flow.

4.1. Preconditions

The fundamental concept of DRAMATICS to establish long-term business relationships is to trial ICT prototypes in real business processes where local partners become co-innovators. This requires that a suitable *business process* be identified before DRAMATICS is applied. The change manager, together with local contacts, conducts contextual analysis and field studies of potential partners to identify processes with a high potential for improvement using ICT. The selection of potential pilot partners and processes requires extensive knowledge of the local context and market situation and is informed by geographical location and distribution, existing organizational structure, willingness to collaborate, and organizational size. Large numbers of or widely dispersed end users can generate significant logistical overhead without benefiting the research. The on-site research results are process maps that inform decisions on a focal process element for the pilot that has been identified as most valuable to be improved with ICT. For example, in the cashew pilot, the ICT solution specifically concentrated on the traceability process from farmer to cooperative, since analysis identified this as a root source of process inefficiencies and as the most valuable process in which to invest.

The team applying DRAMATICS consists of a *core team* plus optional additional team members. The core team, established before commencing DRAMATICS, includes four key roles: project leader, local change manager, local contact, and internal soft-

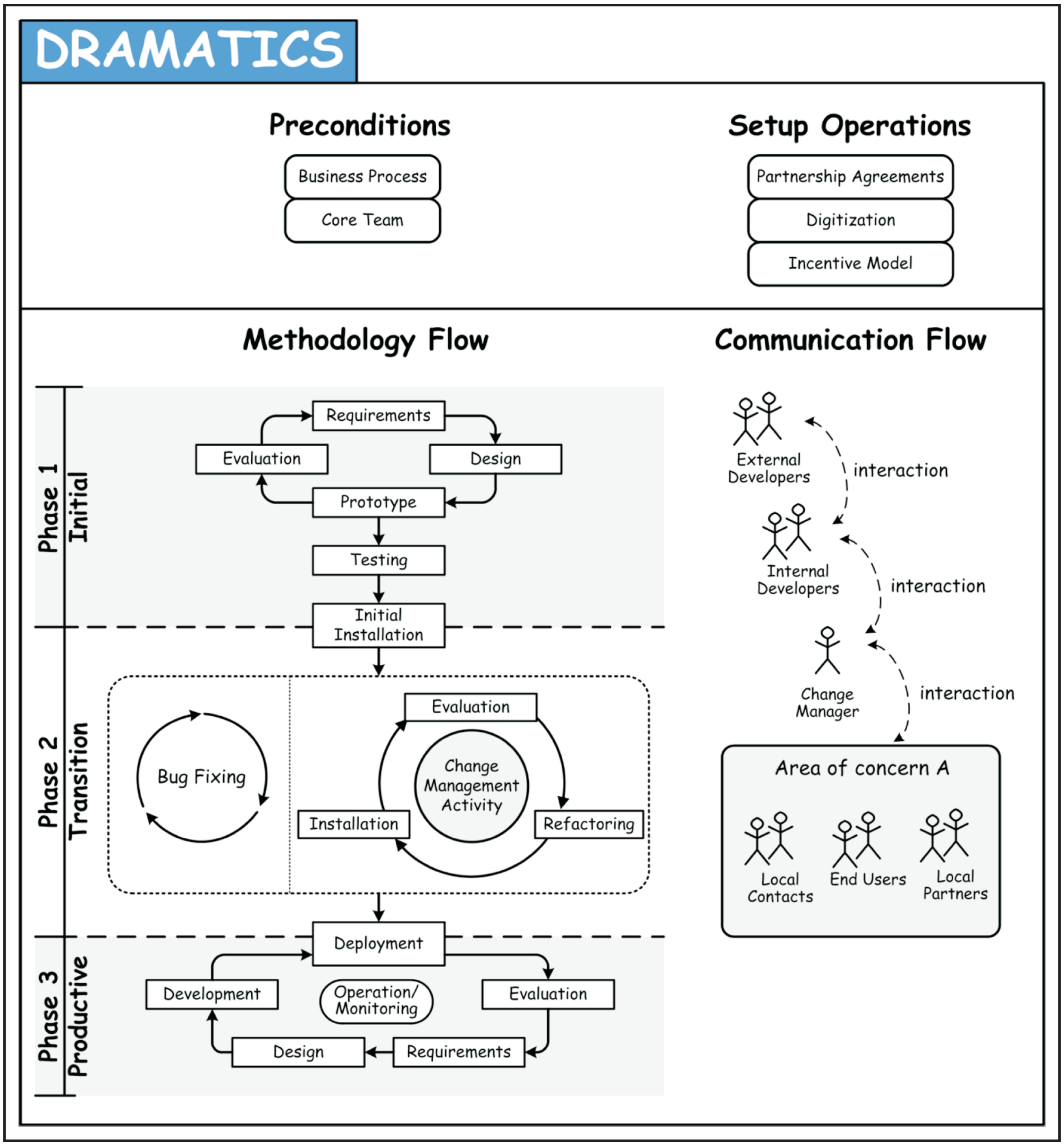

Figure 3. DRAMATICS Overview.

ware developer. Members of the core team require some minimum level of personal reflexivity and exposure to ICTD contexts. Additional team members in supporting roles may be added according to need.

4.2. Setup Operations

Based on a selected focal business process, the change manager sets up *partnership agreements* with local partners to clarify the legal conditions and scope of the collaboration.

After successful negotiations with stakeholders, partner data about users, organizational structures, and processes undergo *digitization.* Ideally this should be the local partners' responsibility, but the core team can provide training material and templates to support data quality.

The *incentive model* defines the compensation to be paid to partners for the additional efforts of trialing prototypes. Incentives have different forms, for example, payment per transaction, per kilo of produce, per registered farmer. The incentive model also clarifies how one-time costs (e.g., phones, SIM cards) and running costs (e.g., airtime) are handled. Even small costs such as SMS charges should be covered since they can break down commitment to a project (Eagle, 2009). When a local partner decides to productively use the ICT system outside of the research project, new collaboration agreements are needed in which the service runs without external incentives.

4.3. Team roles

As with other Agile methodologies (Dubinsky & Hazzan, 2004), the roles used in DRAMATICS (see Figure 3) are an important aspect. We recommend at least one person in each core role, but where resources are limited, it may be possible for one person to fill multiple roles. For larger projects we see no a priori limits to the numbers in the core team, but highlight the importance of a single individual being identified as leading local change management.

The **project leader** is part of the core team. She or he establishes the range of resources available (people, money) within the corporation and is responsible for delivering the overall project goals within budget limits. In our case, the project manager is the corporation's head of research and is the line manager for both the change manager and the internal software developers. The goals are defined in the research project proposal, and the project manager is responsible for delivering results to the project funding organizations. The project leader oversees the entire project but does not directly control its implementation. She or he delegates implementation to the partnership of the change manager (organization and execution of the change in the field including management of the involved local contacts and stakeholders) and the software developers (ICT system development and management of external software developers). The project leader acts as a resource to manage potential breakdowns and conflicts between these two key actors.

The **change manager** is part of the core team and is responsible for all pilot implementation and field activities. She or he is a single design authority for field experiments and ICT solutions. She or he needs substantial on-site experience and soft skills to establish trusting relationships with stakeholders. The change manager bridges the context gap between local stakeholders and internal software developers. Combining technical and business backgrounds, she or he interacts with stakeholders, identifies opportunities, and together with internal software developers, translates them into possible ICT solutions. She or he is the main on-site contact. The change manager leads context analysis, collects feedback and pilot information (e.g., possible collaboration partners, valuable processes), and prepares the information for core team meetings. Organizational (e.g., process decisions, piloting partners) and technical (e.g., UI design, hardware, functional scope) decisions are made by the core team during collaborative meetings.

In DRAMATICS, the change manager is external to the local partner and end user community. This contrasts with Sharpey-Schafer's (2009) "champions," who are local stakeholders with a vested interest in driving the project internally, but who are not necessarily able to assess trade-offs among commercial benefits, technological opportunities, and implementation complexity. The change manager's role corresponds more closely with the role of "development project coordinator" described by Dearden et al. (2010).

In our projects, the change manager is an African of European descent, who grew up in South Africa and Germany and has a background in software and business process design. Although the pilots in Ghana and Burkina Faso are far away from his native South Africa, his understanding of various African cultural contexts is important.

Internal software developers are part of the core team but located outside the use case scenario. They are responsible for delivering the ICT system and coordinating external software developers. The internal software developers must have an understanding of the pilot scenario (from on-site visits or an extensive briefing from the change manager), especially about infrastructure and appropriate hardware. They interact with the change manager to transform design proposals into ICT solutions. They bridge the context gap between the contextual knowledge of the change manager and technical capabilities, including external software developers. As with other Agile methodologies (see, e.g., Beck, 1999), the core of the decision process for each software iteration is the negotiation between the change manager's assessments of potential value in the context and the internal software developers' assessments of technical opportunities and complexities. This negotiation aims to deploy the (restricted) resources for development to deliver the maximum benefit for stakeholders. While the project leader is empowered to overrule team decisions made by the change manager and the internal software developers, in our case he has never done so.

External software developers are optional. They are introduced to the context only at a basic level, using personas and scenarios and through interaction with internal software developers. They execute modular software development tasks that are precisely defined using functional design documents. Systems integration is performed by the internal developers. External developers usually participate in Phase 1 and Phase 2. In productive use, software development workload decreases and is managed by the internal development team.

Local partners sign an agreement to collaborate as co-innovators. In our projects we have two types of contact persons in local partner organizations: in smaller organizations like a cashew union, collaboration is with the chief executive; with larger companies, we collaborate with a responsible person motivated to promote the topic to the higher management level. This role is similar to Sharpey-Schafer's (2009) "champion." In C@R, we also worked with infopreneurs who form an integral part of the new system (e.g., manage Spaza Shop master data, check incoming orders).

Local contacts do not sign a collaboration agreement. They support on-site interactions on a voluntary basis such as assisting with training events, conducting interviews, or collecting data using their local knowledge, personal relationships, and language skills. Examples include volunteer end-users from the community from local partners' staff, or volunteers from international aid organizations.

End-users sign a collaboration agreement, take part as co-innovators, and are compensated using the incentive model. If end-users are employees, then local partners decide which staff are invited to join the project, but research participation is ultimately the decision of the end-users.

Role interactions. To manage communication, DRAMATICS defines a *communication flow* among members of the core team. The primary direct interaction with end users is done by the change manager and local contacts. Internal software developers communicate with the pilot area via the change manager. In some exceptional cases, when the project has matured, internal software developers might discuss software issues with local contacts directly. Internal software developers coordinate the work of external developers. There is no direct communication between external software developers and the pilot stakeholders. The role specialization between change manager and internal software developer is a distinctive element in DRAMATICS compared to many other ICTD methodologies.

4.4. Three-Phase Cyclic Approach

DRAMATICS as currently used has a three-phase iterative approach (see Figure 3): an initial design phase to arrive at robust operational prototypes; a transition phase in which the prototype is deployed and beta tested and end-users are trained; and a productive use phase where the system is used in a live business process. The cycle structures and frequencies are different in these three phases.

Phase 1: Initial Prototyping

In Phase 1 prototypes are developed for the focal business process. The change manager, together with internal software developers, creates a requirements document as input for paper UI mock-ups used in initial (online or face-to-face) workshops with local partners. In several subsequent workshops mock-ups are refined until they adequately cover the selected processes. Mature UI mock-ups and refined processes then inform the first end-user workshop. While discussions of UI mock-ups and processes are an important part of the first workshop, the purpose at this stage is to introduce the research intervention and possible implications to the end-users. In several subsequent workshops with partners and end-users, prototypes are refined (see Figure 4). Since many end-users have difficulty imagining the ICT system using paper prototypes, a rapid shift to functional prototypes is recommended. The final prototypes are tested by available resources of the core team, interns, and external volunteer testers. The results of Phase 1, after several iterative cycles, are robust prototypes, ready for deployment.

Phase 2: Transition

In Phase 2 the robust prototypes are installed for the first time in the real environment (*initial installation*) for testing and training.

Figure 4. Paper-based and functional design workshops executed by change manager and local contacts (ACi).

Figure 5. ICT system training at the workshop facilities and in the field (ACi).

In a series of training workshops, all pilot stakeholders are informed about the adapted business processes, the ICT systems, and the consequences to their day-to-day work (see Figure 5). For training, daily work tasks are simulated (e.g., training to scan a barcode on a jute sack using a smartphone camera), adopting a user-trains-user approach. Preparation for training includes creating documentation, which is translated into local languages and printed in advance since IT infrastructure is unreliable. After the workshops, further training happens at the actual process locations.

Active use by real end users often reveals bugs, which leads to minor system adaptations and new functionality requests. Typically, bug fixing can be distributed to interns and external developers, but system adaptations and new functionality require the contextual knowledge of internal developers. Daily feedback from the field requires a close interaction between the change manager and (remote) internal software developers. To cope with the communication challenges, plans are needed for how change requests are reported and how software updates reach the change manager in the field. The change manager needs the technical skills to direct installation and update tasks, which may imply refresher training prior to this phase.

In our pilot projects, the transition phase took 1–2 weeks. In ACi and SSN, the transition phase was scheduled directly before the productive phase (i.e., the harvest) so that the change manager's last few days on site coincided with the first real transactions on the productive system.

The result of Phase 2 is system deployment in a real business process and active use by trained end users. From this point forward the former prototype is now a productive system with all its demands regarding reliability, usability, and security.

Phase 3: Productive Use

Phase 3 supports productive use of the system. Since updates must be carefully planned in a live system to ensure business continuity, iterative cycle time increases. Logistics costs also play a role if devices need updating with new software. The *operation/monitoring* activity in Phase 3 represents the continuous tasks of monitoring systems and remote support (e.g., regular calls with local contacts, collecting end-user feedback, replenishing mobile airtime).

To control costs during live operation, local support and feedback go through local contacts. From their learning in phases 1 and 2, they can execute much support independently. Phase 3 also includes evaluating the system, preparing to scale up, and adjusting business models for the system after the research project ends.

5. Summative Evidence for the Effectiveness of the Methodology

Key to success measurement in ICT4D research is the positive development impact it generates (Toyama, 2010). Drawing on Heeks and Molla's (2009) concept of the ICT4D value chain (see Figure 6), the DRAMATICS methodology can be seen as an *input* for a development intervention, which supports the *implementation* of both process changes and ICT systems, that is, "intermediates" in

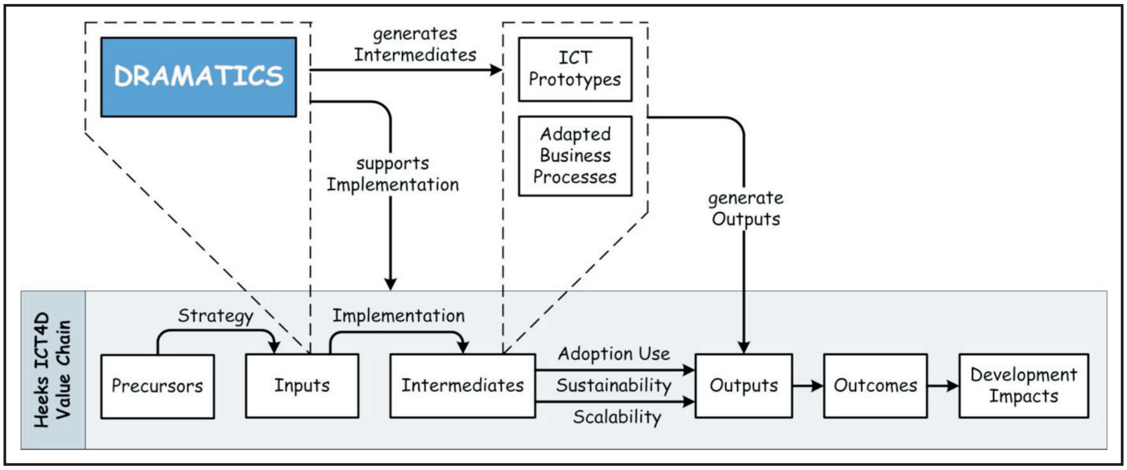

Figure 6. DRAMATICS mapped to Heeks' ICT4D value chain.

Heeks' model. If systems and processes are adopted, sustained, and scaled, they generate development *outputs* (e.g., new communication patterns, more efficient processes). *Outputs* lead to *outcomes,* quantitative and qualitative benefits or disadvantages for different stakeholders and, ultimately, to *development impacts.*

Following this framework, our DRAMATICS evaluation is based on reviewing the *outputs* in the pilot scenarios. For each case study we explain the systems' development effort and the functionality. We present evidence of adoption, explore financial sustainability, and discuss current plans for scaling up. The data presented below have been gathered by change managers, interns, and local contacts through on-site field studies, workshops, questionnaires, personal interviews, direct observations, and system log file analyses from 2007–2013 (Figure 7). As a software engineer and an interactive systems designer, we suggest that evaluation of the wider *outcomes* and ultimate *development impacts* now demands work from researchers in other disciplines.

5.1. Retail: Collaboration@Rural (C@R)

Pilots: After 22 months of software development, the system was deployed for a nine-month pilot in 2009 with 21 Spaza Shops spread across 10 villages, two infopreneurs, and one supplier. After the C@R project ended in August 2010, research continued with a follow-up project that we supported in software development. Since September 2012, the C@R scenario has been funded by a donation from SAP's corporate social responsibility section. In December 2012 we deployed a redesigned ICT system, including a new supplier.

Team: One change manager, two internal software developers, one project manager, 5–10 local contacts (two of them long-term contacts), and 2–3 interns.

ICT System: To improve bargaining power, individual Spaza Shops are organized into "virtual cooperatives." The system allows shopkeepers to restock, placing orders through structured SMS on low-end mobiles (Figure 8, on left), with stock delivered by the supplier. An infopreneur mediates between the shop and the supplier, using a desktop application (Figure 8, on right) to validate and bundle orders.

Support: Technical support for the pilot was done by the research team. The involvement in the research activities together with the researchers enhanced the infopreneurs' skills. Those enhanced technical and teamwork skills then enabled them to handle certain on-site activities on their own, enabling them to handle registration of new Spaza Shops, end-user support, order management, and organization of product delivery. System maintenance and software updates are handled by the research team.

System Adoption: During the pilot 77% of the shops actively used the SMS ordering system. The number of participating shops has increased from 21 in May 2009 to 28 in 2012.

Sustainability: The *supplier* benefits from more efficient demand and logistics planning, avoiding unfulfilled customer demand or returned goods. Sellers have reported a 2% sales increase. For the

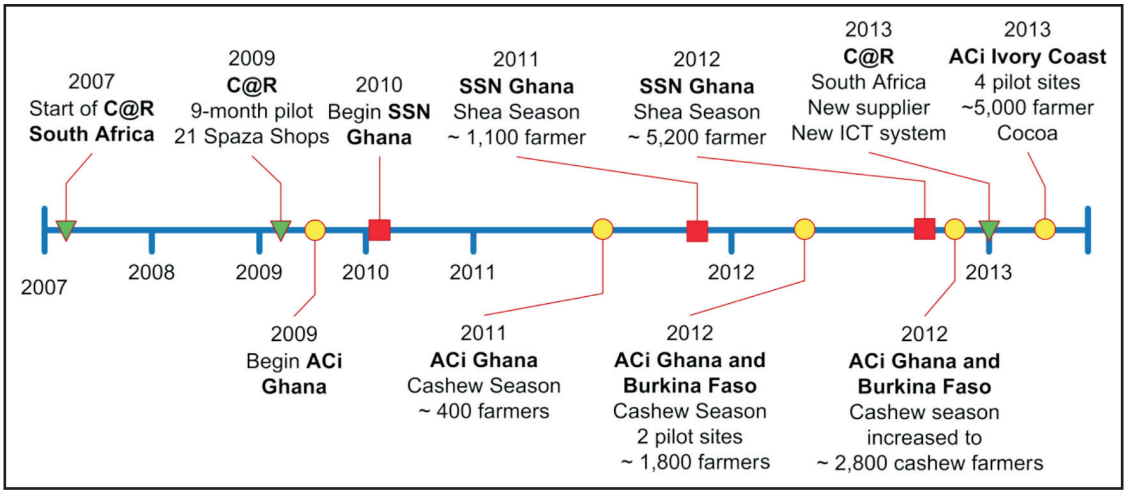

Figure 7. Coarse-grained timeline of the three projects, C@R, SSN, and ACi.

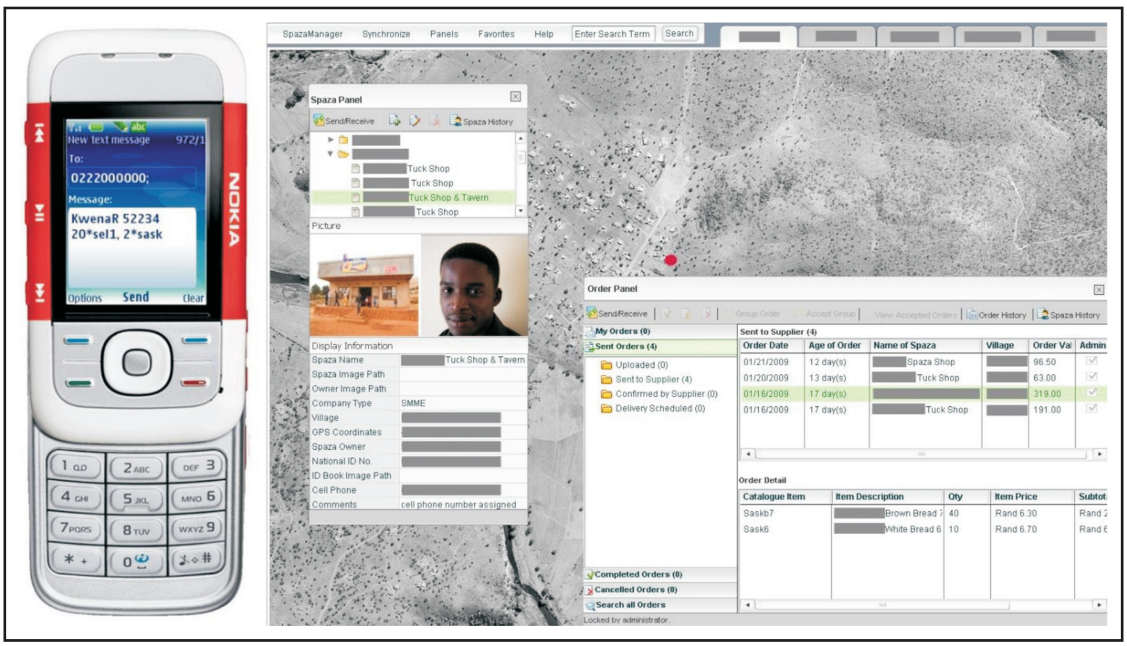

Figure 8. SMS application (left) and Infopreneur desktop application (right, anonymized).

shops, "at-the-door" delivery and lower bulk order prices cut procurement overhead from a previous US$125 to approximately US$12.50 per month. Today *Spaza Shops* pay a transaction fee of roughly US$5 per delivery, leaving a net benefit of US$107.50 per *shop* per delivery. With 30 shops, this results in US$3,225 savings per month that cover

- an operational cost of US$375 per month for the *infopreneur,* currently compensated from the research budget, who receives a monthly salary of US$250, which is competitive for high-potential individuals in the informal economy;
- US$50 monthly server rental;
- US$0.50 SMS costs per Spaza per month;

Figure 9. Smartphone application used in the field (left) and union office desktop application (right).

- US$10 mobile Internet costs for the desktop application; and
- US$50/month to finance the infopreneurs' laptops (US$1,200) within two years.

This leaves a total revenue gain of US$2,850 for the 30 shops.

Scaling Up: Starting with 21 shops in 2009, today 28 shops regularly place orders. Since mid-2012 a larger supplier has joined as local partner, providing more products. External support is currently 1–2 person-hours per week to monitor server status. Operational costs are still covered by a donation fund, but the goal is to transform the pilot into a sustainable social business by early 2014.

5.2. Agriculture: African Cashew Initiative (ACi)

Pilots: After 20 months' development the ICT was piloted with the 2011 cashew havest. With approximately 400 participating farmers, roughly 120 tons of raw cashews were processed via the system. Pilots scaled up to about 2,800 cashew farmers at the end of 2012.

Team: One change manager, three internal software developers, 4–5 external software developers, one project manager, 10–20 local contacts, 3–4 interns.

ICT System: Farmers' membership booklets and cashew jute sacks have barcodes attached that are scanned using strengthened offline-capable smartphones at buying stations (Figure 9, on left). At the central cashew union office, an (occasionally) online desktop application enables the management and analysis of transaction data from the individual stations (Figure 9, on right). To cope with an unreliable power supply, especially at buying stations in remote villages, the smartphone application and setup are configured for minimum energy consumption, and a spare battery is included in the initial smarthphone distribution.

Support: The research team carried out operational support for the pilots in Ghana and Burkina Faso with the assistance of continuously trained local contacts. The research team continues to manage the hardware and back-end systems.

System Adoption: In 2011, only 4% of all cashew purchases at participating buying stations bypassed the ICT system, mainly because of detached barcode tags. In interviews with 105 farmers, they expressed the following benefits: exact record keeping, transaction data analytics, improved visibility, increased trust in the union books, reduced input errors, and better logistics planning.

Sustainability: The ICT system enables the cooperative to secure a Fair Trade premium of approximately US$0.30 per kg, generating approximately a US$7,200 premium[2] per harvest. Operational costs including server hosting, airtime, barcode tags, and incentive model are approximately US$4,000 per harvest and are covered by this premium. Hardware (10 smartphones and one laptop) at approximately US$4,800 can be paid for in only two seasons by the remaining premium. A side benefit is farmers' increased trust in the cooperative management, attracting more farmers. With the cooperatives' increased size, relative ICT costs will

2. 120,000 kg of raw cashew nuts result in approximately 24,000 kg of cashew kernels after processing (shelling, drying): US$0.30 x 24,000 kg = US$7,200.

decrease. All financial gains benefit famers directly, since fair trading means they must be paid a dedicated percentage of the cashew trade price.

Scaling Up: Starting in April 2011 with approximately 400 farmers from one cooperative, the system has scaled up to approximately 2,800 cashew farmers from three cooperatives in October 2012. The successful pilots have attracted further crop processors and traders who participated in additional pilots in 2013, including about 5,000 cocoa farmers in the Ivory Coast.

5.3. Agriculture: Star Shea Network (SSN)

Pilots: After an initial on-site visit in December 2009, iterative development took place from June to September 2011. The ICT system was deployed in a shea value chain in 2011 with more than 1,100 actively participating women (approximately 3,000 had registered for the program) of the Star Shea Network (SSN), a national organization of small cooperatives in northeast Ghana. Roughly 170 tons of shea kernels were processed by the ICT system in 2011. The pilot grew to about 5,200 farmers in 2012.

Team: One change manager, three internal software developers, one project manager, 5–10 local contacts, 3–4 interns.

ICT System: The ICT system is similar to the cashew solution, with smartphones at buying stations tracing shea purchases. At the union office the desktop client is used for master data management and transaction data analysis. The same mechanism manages data synchronization between devices.

Support: Operational support was realized through a collaboration of the research team and local contacts. Similar to the cashew pilots, the research team provided technical support such as server maintenance, while local contacts provided end-user support and software update installations.

System Adoption: A system adoption rate of 97% and 100% of transactions and a transaction accuracy of 95% and 100% were achieved in the pilot 2011 and 2012, respectively.

Sustainability: System sustainability and financial benefits for the pilot stakeholders are similar to the cashew project. However, in the shea pilot, the higher total number of participants and higher number of participants per station (implying fewer smartphones) make the finances even better. Stanford University has conducted an independent evaluation, which noted that participating women have been able to increase their incomes by between 59% and 82% (Rammohan, 2010).

Scaling Up: Starting with approximately 1,100 farmers in September 2011, the system scaled up to approximately 5,200 farmers in September 2012. In 2012 the SSN transformed from the research setup to the social business StarShea Ltd. To attract more partners along the value chain, new functionalities such as pre-finance and payments were trialed during the 2012 harvest. In 2013 the pilot will grow to about 10,000–15,000 farmers.

6. Conclusion

We have shown in detail the structured approach taken to iteratively refine the DRAMATICS concepts to its current state. We elaborated on the details of DRAMATICS and provided evidence of its effectiveness through its application in research pilots. We are now at a stage where additional independent evaluations, similar to the Stanford University study of the shea pilot (Rammohan, 2010) would be appropriate. In particular, given the tight focus of these projects on creating sustainable commercial solutions to support livelihoods, it would be valuable to explore some of the broader project outcomes, for example, how participation has affected the capabilities of organizations and communities with which the project has worked.

While preparing this article, DRAMATICS was instantiated in additional shea and cashew pilots in Ghana and Burkina Faso as well as new pilots in the Ivory Coast's cocoa sector. Due to the similar processes and pilot structure, an improved ICT system, and DRAMATICS' well-structured processes, the time from first UI mock-ups to productive use of an ICT prototype in a pilot scenario was reduced to two weeks, and on-site activities now are mainly driven by local contacts with the change manager being on-site only one-to-two days over the course of the whole development. Current activities are focused on expanding pilots to 30,000 farmers by the end of 2013 and instantiating new models for support as sustainable, independent business entities. We are also preparing to use DRAMATICS to develop solutions for other agricultural sectors such as coffee and cotton.

These results and evaluations show DRAMATICS to be a repeatable software development methodology, enabling the creation of sustainable, scalable, and reusable ICTD solutions in multiple contexts. Every software development project is unique in some respects, and (reflecting the interests of the corporation) the projects where DRAMATICS has been applied share a common theme of supply chain management in the private sector. Thus, DRAMATICS is not a recipe to be followed blindly. Rather we offer this account of DRAMATICS as a resource that others can appropriate and adapt to fit their particular project contexts. At the core of DRAMATICS' philosophy is the tight collaboration between a situation-focused but technically-aware change manager working closely with partners and end-users on site and the technically-focused, but situationally-aware software developers. This approach has demonstrated its effectiveness for supply chain operations in the retail and agricultural sectors. With application and evaluation in other sectors such as education, health, transport, etc., we hope that DRAMATICS can be developed and extended toward a dependable family of approaches that can be applied in different ICTD contexts. To this end we welcome further instantiations and evaluations of DRAMATICS to improve our understanding of effective ICTD software development strategies. ■

References

Anokwa, Y., Ribeka, N., Parikh, T., Borriello, G., & Were, M. C. (2012). Design of a phone-based clinical decision support system for resource-limited settings. *Proceedings of the 5th International Conference on Information and Communication Technologies and Development (ICTD'12)* Atlanta, GA (pp. 13–24). doi:10.1145/2160673.2160676

Baskerville, R., & Myers, M. D. (2004). Action research in information systems: Making IS research relevant to practice-foreword [Special issue]. *MIS Quarterly 28*(3), 329–335.

Beck, K. (1999). *Extreme programming explained: Embrace change.* Reading, MA: Addison-Wesley.

Blake, E. (2010). Software engineering in developing communities. *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE'10)* (pp. 1–4). Capetown, South Africa.

Blake, E., & Tucker, W. (2006). Socially aware software engineering for the developing world. In P. Cunningham & M. Cunningham (Eds.), *IST Africa 2006* (pp. 1–8). Pretoria, South Africa.

Brewer, E., Demmer, M., Du, B., Ho, M., Kam, M., Nedevschi, S., et al. (2005). The case for technology in developing regions. *Computer, 38*(6), 25–38.

Byrne, E. (2003). Development through communicative action and information system design: A case study from South Africa. *Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists* (pp. 83–92).Fourways, South Africa.

Checkland, P., & Holwell, S. (1998). Action research: Its nature and validity. *Systemic Practice and Action Research, 11*(1), 9–21.

Dearden, A., & Rizvi, S. M. H. (2009). A deeply embedded sociotechnical strategy for designing ICT for development. *International Journal of Sociotechnology and Knowledge Development, 1*(4), 52–70.

Dearden, A., Rizvi, H., & Gupta, S. (2010). Roles and responsibilities in agile ICT for development. In A. Joshi & A. Dearden (Eds.), *Proceedings of India HCI 2010/Interaction Design for International Development,* Mumbai, India, 19–30. Retrieved from http://ewic.bcs.org/content/ConWebDoc/35770

Doerflinger, J., & Gross, T. (2012). Sustainable ICT in agricultural value chains. *IT Professional, 14*(4), 48–55.

Dubinsky, Y., & Hazzan, O. (2004). Roles in agile software development teams. *5th International Conference on Extreme Programming and Agile Processes in Software Engineering* (pp. 157–165). Garmisch-Partenkirchen, Germany.

Eagle, N. (2009). Txteagle: Mobile crowdsourcing. In N. Aykin (Eds.), *Internationalization, Design and Global Development, 5623,* 447–456. doi:10.1007/978-3-642-02767-3_50

European Commission. (2009, January). *Living labs for user-driven open innovation: An overview of the living labs methodology, activities and achievements.* European Commission, Information Society and Media. doi:10.2759/34481

Hartung, C., Lerer, A., Anokwa, Y., Tseng, C., Brunette, W., & Borriello, G. (2010). Open data kit. *Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development (ICTD'10)* (pp. 1–12). London, UK.

Hayes, G. R. (2011). The relationship of action research to human-computer interaction. *ACM Transactions on Computer-Human Interaction, 18*(3), 1–20. doi:10.1145/1993060.1993065

Heeks, R. (2008). 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. (2009). *The ICT4D 2.0 Manifesto: Where next for ICTs and international development?* (Development Informatics Working Paper Series, 42). Retrieved from University of Manchester website: http://www.sed.manchester.ac.uk/idpm/research/publications/wp/di/di_wp42.htm

Heeks, R., & Molla, A. (2009). *Impact assessment of ICT-for-development projects: A compendium of approaches.* (Development Informatics Working Paper Series, 36). Retrieved from University of Manchester website: http://www.sed.manchester.ac.uk/idpm/research/publications/wp/di/di_wp36.htm

Jayaratna, N. (1994). *Understanding and evaluating methodologies: NIMSAD, a systematic framework.* New York, NY: McGraw-Hill.

Kumar, R. (2004). eChoupals: A study on the financial sustainability of village Internet centers in rural Madhya Pradesh. *Information Technologies & International Development, 2*(1), 45–73.

Ligthelm, A. (2002). *Characteristics of Spaza retailers: Evidence from a national survey.* (Research Report No. 305). Pretoria, South Africa: Bureau of Market Research, University of South Africa.

Liu, A. T., & Payne, J. (2011). Software platforms for mobile applications for agriculture development. USAID. Retrieved from https://communities.usaidallnet.gov/ictforag/node/1650

Maunder, A., Marsden, G., Gruijters, D., & Blake, E. (2007). Designing interactive systems for the developing world: Reflections on user-centred design. *International Conference on Information and Communication Technologies and Development* (pp. 1–8). Bangalore, India. doi:10.1109/ICTD.2007.4937419

Merz, C., De Louw, R., & Ullrich, N. (2007). Collaborative working environments for enterprise incubation: The Sekhukhune Rural Living Lab. *Proceedings of IST-Africa'07* (pp. 1–9). Maputo, Mozambique. Retrieved from http://www.c-rural.eu/Southafrica-LivingLab/images/stories/Files/ISTAfricaPaperCWEEnterprises.pdf

Parikh, T. S., & Lazowska, E. D. (2006). Designing an architecture for delivering mobile information services to the rural developing world. *Proceedings of the 15th International Conference on World Wide Web (WWW'06)* (pp. 791–800). Edinburgh, Scotland.

Rammohan, S. (2010). The shea value chain reinforcement initiative in Ghana. *Case Study, Stanford Global Supply Chain Management Forum,* 13. Retrieved from http://www.planetfinancegroup.org/en/publications/research-papers

Sharpey-Schafer, K. (2009). Key techniques in building effective ICT solutions in the South African HIV/AIDS sector. *CCC Workshop on Computer Science and Global Development* (pp. 77–78). Berkeley, CA.

Thinyane, M., & Terzoli, A. (2009). Universal digital inclusion: Beyond connectivity, affordability and capability. *Innovations for Digital Inclusions. ITU-T Kaleidoscope Academic Conference* (pp. 1–8). Mar del Plata, Argentina.

Toyama, K. (2010). Human–computer interaction and global development. *Foundations and Trends in Human–Computer Interaction, 4*(1), 1–79.

van Rensburg, J., Veldsman, A., & Jenkins, M. (2008). From technologists to social enterprise developers: Our journey as 'ICT for development' practitioners in Southern Africa. *Information Technology for Development, 14*(1), 76–89. doi:10.1002/itdj.20088

Veeraraghavan, R., Yasodhar, N., & Toyama, K. (2007). Warana unwired: Replacing PCs with mobile phones in a rural sugarcane cooperative. *2007 International Conference on Information and Communication Technologies and Development, 5* (pp. 1–10). Bangalore, India.

West, D., & Stansfield, M. H. (2001). Structuring action and reflection in information systems action research studies using Checkland's FMA model. *Systemic Practice and Action Research, 14*(3), 251–281. doi:10.1023/A:1011355214452

Winschiers, H. (2006). The challenges of participatory design in an intercultural context: Designing for usability in Namibia. *Proceedings of the Ninth Participatory Design Conference, Vol-II* (pp. 73–76). Trento, Italy.

Zainudeen, A., Samarajiva, R., & Sivapragasam, N. (2011). CellBazaar: Enabling m-commerce in Bangladesh. *Information Technologies & International Development, 7*(3), 61–76.