

Roles and responsibilities in agile ICT for development

DEARDEN, Andy <<http://orcid.org/0000-0002-5706-5978>>, RIZVI, Haider and GUPTA, Subodh

Available from Sheffield Hallam University Research Archive (SHURA) at:

<http://shura.shu.ac.uk/3549/>

This document is the author deposited version. You are advised to consult the publisher's version if you wish to cite from it.

Published version

DEARDEN, Andy, RIZVI, Haider and GUPTA, Subodh (2010). Roles and responsibilities in agile ICT for development. *Electronic Workshops in Computing*, 1-12.

Copyright and re-use policy

See <http://shura.shu.ac.uk/information.html>

Roles and Responsibilities in agile ICT for Development

Andy Dearden
Communication & Computing
Research Centre,
Sheffield Hallam University, UK
a.m.dearden@shu.ac.uk

Haider Rizvi
School of Good Governance and
Policy Analysis, An autonomous
institution of the Government of
Madhya Pradesh, Bhopal India
smhaiderrizvi@rediffmail.com

Subodh Gupta
Safal Solutions Private Ltd.
Secunderabad, Andhra Pradesh,
India.
subodh.safal@gmail.com

This paper examines the different roles in designing interactive software in a ICT for development context. Using experiences from a participatory action research project, in which we used agile methods to design and deploy a system to support ‘agricultural information flow’ for a co-operative of small farmers in rural India, we identify points of difference between the roles in standard descriptions of agile software methods and the roles as they emerged in our project. A key finding is the critical role played by a ‘Development Project Manager’ in facilitating dialogue, orchestrating the activities of other actors and in building the capabilities and confidence of all the participants in joint action.

Keywords: Action Research, Agile Software Design, ICT for Development, Roles.

1. INTRODUCTION

In this paper, we consider different ways of configuring and distributing roles in a participatory project to create new interactive systems in a social development context. We review our experiences in a participatory action research project, that has designed, developed and deployed an ‘agricultural information flow system’ in collaboration with a co-operatively owned crop producers’ company in Sironj, Madhya Pradesh. We examine the roles described agile software development methodologies, and review how these roles evolved in our project. Our findings reflect how the context of interaction design for development differs from other forms of software production, and suggest alternative perspectives on roles in this setting. In particular, we identify the critical role in our project of the ‘Development Project Co-ordinator’, and we recommend that this role should be recognised and supported by future projects.

Structure of this paper

In the next section we give an outline of the project in which our work has been conducted. Section 3 discusses the framework of action research that we have applied. Section 4 examines different divisions of roles in agile software development methods. Section 5 presents the main analysis of the incidents within the project on which we base our findings. Section 6 discusses the implications of our experience for future projects.

2. BACKGROUND TO THE PROJECT

It is not the aim of this paper to discuss details of the software and the basis for technical decisions made in designing, our emphasis is on the organisation of the designing activity itself. However, it is helpful to provide some background to situate the discussion that follows.

Project aims and funding

The central research aim of the project has been to “explore how techniques from the fields of participatory design of Information & Communications Technologies (ICT), agile ICT development and participatory rural appraisal can be combined to support the (locally based) development of sustainable software and business systems for use by networks of rural village co-operatives.” (Dearden et al., 2006). Our approach has combined literature and project reviews with an action research intervention designing a software system in one particular setting. The project was funded by the UK Engineering & Physical Sciences Research Council as part of an initiative on “Bridging the Global Digital Divide”. It was a multidisciplinary research project, bringing together researchers in software design, ICT for development, interaction design, business models in economic development, and an Indian software house that specializes in

providing solutions for the co-operative and NGO sectors.

Project partners

The primary internal partners in the part of the work described here have been: a researcher in participatory methods, working as an employee of a UK University, a principal investigator at the university who has overall responsibility for the project; and the Indian software development house. The goals of these partners were to find ways to combine participatory methods in ICT design and participatory methods in social development, to design an information system for use in Rural areas that could deliver multiple e-services. The external partners (who were identified and selected after the project began), are a large Indian NGO, Pradan, that works to facilitate development programmes by providing professional workers to work with communities to build up their capabilities and to support community based organisations, and the Sironj Crop Producers' Company Ltd. (SCPCL), an agricultural co-operative that is supported by the Madhya Pradesh government's District Poverty Intervention Programme (DPIP). SCPCL has approximately 500 members, each of whom must purchase at least one share in the company, which they must purchase for 100 Indian Rupees (approximately US\$2.20). The members are located in many different villages across the Sironj Block varying from 5 to 30Km from Sironj town.

The system being designed

Based on discussions with the DPIP, Pradan and SCPCL, the project focused on supporting the 'Agricultural Information Flow System'. The primary goal is to improve communications between SCPCL's agricultural advisor, who works mostly from the co-op offices in Sironj and makes visits to the different villages to discuss problems and disseminate advice, with the farmers in the villages.

The system allows telephone conversations between the advisor and farmers to be recorded, indexed and then accessed using an Interactive Voice Response System. It also allows audio-visual messages, combining up to six digital photographs together with an audio track, to be exchanged. These audio-visual messages can be created using camera phones which have been provided to a 'service providers' or 'Munnas'¹ in each village. The system has required the development of bespoke software to run on mobile phones (developed using the Python interpreter running on Symbian 60 mobile phones) and web server software running on a dedicated PC server. The basic structure of the audio-visual messages is strongly influenced by the software developed in the Storybank project (Jones et al., 2008).

Using these enhanced communications, it is hoped that the advisor will be able to respond more rapidly to farmers' queries (currently farmers may wait a few days before the advisor can visit their village); to deal more easily with 'frequently asked questions'; and will be able to distribute general information (e.g. weather information, upcoming events etc.) more easily.

RESEARCH METHODS

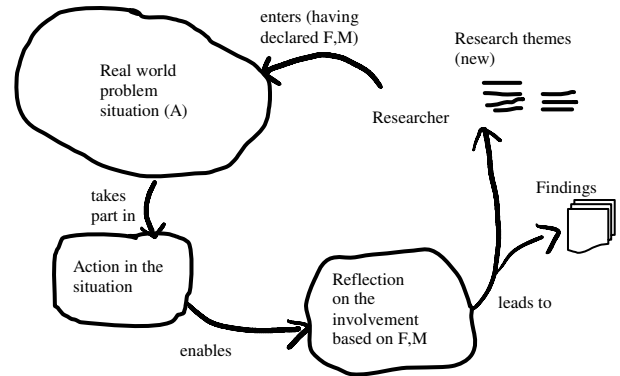


Figure 1: The cycle of action research in human situations. Adapted from Checkland & Holwell, 1997, p15

This project is an action research enquiry into participatory and agile software methods, conducted in the context of designing new interactive systems for a rural co-operative. Action research is an established research approach in information systems (see Baskerville & Myers, 2004), and there is a diversity of action research methods reported in the literature (Baskerville & Wood-Harper, 1998). As Checkland & Holwell (1997) discuss, the validity of claims made from action research studies cannot achieve the kind of replicability that is characteristic of the empiricist methods of the physical sciences. Rather, they argue that researchers should seek to enable 'recoverability' of the research activity, so that a reader of the work is able to identify starting assumptions, assess potential biases in interpretation, and the likely applicability of the findings in another context. In particular, the framing assumptions of the research must be declared at the beginning of the research cycle, and opening up discussions of interpretation to the subjects / participants in the research process.

In this project, we follow Checkland & Holwell's (1997) framework. That is, we wish to investigate a specific methodology M, in a specific real world problem situation A and we have a framework of ideas F that is guiding our intervention. In order to ensure reasonable 'recoverability', Checkland & Holwell demand that the methodology, the existing framework of ideas and the guiding assumptions be made explicit at the beginning of the research process. The researcher then enters the problem situation, engages in an iterative, participatory action process, and reflects upon the processes to record learning about F, M, and A. The

¹ The name Munna was derived in a workshop and refers to the popular Hindi film character Munna Bhai.

outcome of this process cannot be the refutation of a hypothesis (as with empirical investigations in the physical sciences), but is more often a set of findings specific to the context of application, and the development of research themes. Figure 1 illustrates Checkland & Holwell's recommended process.

In this project, the problem situation (A) was designing software for a rural agricultural co-operative in India. The methodology (M) was an effort to combine agile software development methods, specifically we focussed on Beck's (2005) eXtreme Programming, with participatory design methods in IT and participatory rural appraisal (See Dearden & Rizvi, 2008 for a review). The framework of ideas focused on both: the software design process described by Beck in terms of rapid iterative development cycles and the distribution of roles within the project based on descriptions in Beck (2005) and in reports of other agile methods. We assumed that the iterative structure of agile methods would support the iterative participatory design of interactive software, and would be compatible with participatory social development approaches.

One distinction that should be noted is that agile methodologies typically expect the software developers and representatives of users and customers to be co-located during development. In the case of Indian farmers, this is not feasible. The software developers are not able to work in rural areas where electricity and internet connectivity are either unavailable or very unreliable, but the farmers are not able to leave their fields for weeks at a time to participate in software development.

The findings below have been generated through discussion and reflection upon particular actions and incidents in the project. These incidents and actions were often associated with exchanges of emails and documents. The incidents and actions were discussed and analysed by the current authors at the time (or within a few days of the event) and brief notes written. They have also been discussed and reviewed with other stakeholders in preparing this paper. For these reasons, we expect that our findings will be of interest and value to other researchers working on similar initiatives.

ROLES IN AGILE METHODS

Dubinsky & Hazzan (2004) and Abrahamsson et al. (2002) provide comparative reviews of different role allocations that are used in common agile methods. Some methods provide more fine grained divisions of different roles than others. For example Scrum identifies three major roles in the development organization: scrum master, product owner, scrum team member; together with customer and user as roles in the customer organization and finally the management of the software development organisation. The product owner handles communication with customers and users and sets priorities. The scrum master is a project manager who

supports the team to ensure that resources are available when needed, that obstacles to progress are removed, and that agile principles are followed. The scrum team members conduct the work including all development & testing tasks (Abrahamsson et al., 2002). Beck's (2000) description of Extreme Programming (XP) identifies seven roles, increasing to eleven in his more recent description (Beck, 2005) and Dubinsky & Hazzan's (2004) analysis of Dynamic Systems Development Method (DSDM) lists eleven distinct roles.

These methods do not require that each role is filled by a distinct individual, but they highlight particular tasks and responsibilities that need to be performed. An XP team includes programmers, customers, testers, trackers (who monitor progress), a coach (who has overall process responsibility), as well as consultants and a boss who are viewed as outside the team. DSDM distinguishes four different roles for users and customers, namely: an executive sponsor who acts as project champion at a in management and commits funds and resources for the project; an ambassador user, who represents the user community; a visionary user, who holds the vision of the product; and an advisor user who brings daily business knowledge to the software team. Crystal Clear identifies roles of 'sponsor', 'user' and 'business expert'.

As well as identifying roles for people from within the customer organisation, some agile methods allocate specific roles for managing communications and co-ordination between the customer organisation and the software development team. XP requires that the tester works from the customer's viewpoint and identifies roles for project managers to "facilitate communication", product managers to "encourage communication between customers and programmers", for interaction designers, and technical writers to create "closer relationships with users". Scrum has the 'product owner' who interacts with customers and users and prioritizes the functionality that is developed by the scrum team in each iteration. The product owner corresponds to the 'visionary' in DSDM. On the basis that typically the software delivery organization will want to develop software that can be sold to multiple customers, the product owner is usually (though not always) an employee of the software development organization. DSDM and Adaptive Software Development both identify the role of facilitator with responsibilities for planning, managing and facilitating design workshops and decision making sessions.

Naturally, each agile approach also includes specific roles for technical staff to produce the software and related outputs. These roles include: programmers in XP; senior developers and developers in DSDM; the senior designer, designer-programmers, co-ordinators, testers and writers in Crystal Clear; chief architect, chief programmers and class owners in Feature Driven Development (FDD); master developers and expertise leaders in Lean development; developers in adaptive software development; and scrum team members in scrum.

Scrum	XP	DSDM	Feature driven development	Crystal Clear	Adaptive Systems Development	Roles at start of Project
		Executive sponsor		Sponsor	Executive sponsor	Sponsor (Principal Investigator)
Product owner		Visionary user	Chief architect	Senior Designer		Visionary user / Product owner (Researcher)
	Customer	Ambassador user	Domain experts	User Business expert	Customer	Users (SCPCL members)
Scrum master	Coach Tracker	Project manager Scribe	Project manager	Co-ordinator	Project manager	Project manager (MD of software house)
Manager		Team leader				
Scrum team	Programmer Tester	Technical co-ordinator Senior developer Developer	Development-manager Chief-programmer Class owners	Designer-programmer Tester Writer	Developer	Development team (Employees of software house)
					Facilitator	Facilitator (Researcher)

Comparison of Roles: Based on role descriptions in Dubinsky & Hassan, 2004.

Most advocates of agile approaches accept the need to adapt roles and practices to the specifics of local conditions (Nodder & Neilsen, 2009). However, it is clear from the above that the roles identified in agile methods generally fall into a small number of general classes. There is a common recognition of the need for a project *sponsor* or champion in the customer organization. Additionally, most methods recognise an explicit distinction between project sponsors (usually senior managers) and the *users* of the system. Within the software delivery organisation, there is a role for a *project manager* to co-ordinate activity, to track project progress, encourage communication and ensure that necessary resources are available. There is a need for technical *developers*. In most methods, there is an explicit roles for a *facilitator* to support good communication between the customers, users and developers, and to help the team to reach agreement about priorities in each development cycle. Finally, most methods recognize the need for a single point of design authority, a *product owner*, who owns the overall vision of the product.

Interaction design and agile development

A number of authors have explored how user centred interaction design can be integrated with agile software development practices (Chamberlain, Sharp & Maiden, 2006; Lee & McCrickard, 2007; Ferriera, Noble & Biddle, 2007; Nodder & Neilsen, 2009). A common finding is that some overall conceptual design needs to take place up front before coding starts, and that during the development, the UI design team needs to work at least one cycle ahead of the software development team to ensure that good interaction design proposals are available for the developers when they start

building each piece of functionality. In this project, there was no specialist interaction design resource available. Our solution was that, at the start of each software iteration, a participatory design meeting was held with the farmers and the software developers to create paper prototypes of screen designs. These were then taken as input to the software design.

ROLE EVOLUTION IN THE PROJECT

In this section, we reflect on our experiences in understanding the roles as the project has progressed. Our findings are illustrated by analysis of specific incidents that occurred, and demonstrate how the various roles required re-negotiation and re-definition to suit our rural development context.

Initial roles in the Rural e-Services project

The structure of the funding scheme, as a research project funded by a UK research council, meant that the primary overall responsibility for the project had to be held by an academic with a permanent contract at a UK university (or research centre), and is given the title 'principal investigator'. Our initial understanding was that the principal investigator began with a role similar to the agile project *sponsor*. The staff and members of SCPCL were clearly in the role of *users*, and the staff of the software development house were therefore in roles related to *developers*, including team leaders, technical co-ordinators, software architects etc. The *project manager* for the software development was the managing director of the software house. The researcher undertaking the project field work, working directly with SCPCL and other agencies in Sironj and Madhya Pradesh (MP) was an experienced

development professional and technology project manager, but was not a software developer, and we interpreted his primary role as one of *facilitator*, with a secondary role as the *product owner* or *visionary user*. Because of the limited resources available to the project, we did not have an experienced interaction designer who could participate in the design sessions. The principal investigator was the only team member with relevant experience, and he was located in the UK, not in India.

Enabling the role of user and customer

In agile methods one major task is to write 'user stories' or 'customer stories' that form the basis for defining software features. These stories are short descriptions of 'units of customer-visible functionality' (Beck, 2005, p44), written in a couple of sentences on small cards. These stories describe functionality that users will find valuable, and allow software designers to estimate the effort needed to implement each idea. The customers and users can then decide which stories to include in the next software cycle. The idea of defining a software system using stories, written from the end users' perspective, is a common feature of many approaches to software design (Preece et al. 2004). However, there are significant differences between users stories in XP and the richer scenarios used in interaction design (see Nodder & Neilsen, 2009, for a discussion).

To begin systems design we organised a first story telling workshop to collect stories about how agricultural information was shared in the community, and to imagine how this might change. This one day workshop was attended by 22 ordinary co-op members, the 8 directors and the 6 employees of SCPCL, representatives from Pradan, the researcher, and the project manager from the software development company. Prior to the workshop, smaller meetings between the facilitator and groups of farmers had been held to discuss information flow in the co-operative. During the workshop, participants worked in groups to imagine stories of how they might use new technology. Although the users were familiar with mobile phones, and had seen camera phones, they found it difficult to recall and articulate specific conversations about agricultural information exchanges, or to envisage stories about how these could be in future. Rather than persist with an activity that did not make sense to the farmers, we encouraged them to develop general stories about information and knowledge in their farming. Some example stories that were told are given below:

I am *Pappu*² from the *Kamlapur* village. I sowed Soybean in 2 hectare with required quantity of seeds/bigha. It rained and only ¼ of the field germinated. It was a very dire situation and big loss. I don't what I should have done.

I have got 1.5 hectare of land, I come from *Gulabganj* village and my name is *Guddu*. I sowed *Urad* (a kind of pulse) in June. In the early days the plants were not of good quality but as these grew it formed good shape but even then it didn't give fruits/produce. I was not having any clue on this and was not having any mechanism to get timely information on this.

My name is *Pinki* and I belong to the *Surajmukhi* village. We have 1.5 hectare of land. Our community was a nomad and we are the first settlers. We were very poor and suffered a lot and nobody gave us any value and space in the society. We were working as daily wage labourers and used to sell jungle woods. At the same time we were away from the information and its sources. It was only when we got associated with Pradan we realise the importance of organisation as we formed a SHG³. We were working on our fields but were not getting any return. We used to sow 25 kilograms of Soybean in one *bigha* of land and were getting only 50 kilograms as the produce. It was after getting associated with Pradan and SCPCL that we come to know the problems in our soil and how to correct that. We also learnt about the methods of sowing e.g. 18 inches sowing. Now we are getting 7 to 12 quintal in a bigha. We believe that the information provided to us played an important in our prosperity and empowerment.

Whilst the story telling was valuable in building commitment in SCPCL, these stories were too general to be usable as inputs to software design. We needed to find new methods to help the farmers to structure their stories at a suitable level of detail.

Our solution involved creating a set of personas (Cooper, 1999) to represent the primary actors in the communication activities. These represented: farmers of different socio-economic orientations and land holdings, the 'service provider' (or Munna⁴) who works for the co-op and holds the mobile phone, the agricultural advisor, and external experts. Taking these personae, a member of the software development team drew a set of pencil sketches to represent each of the personae on a single sheet of A4 paper. This was then photocopied so that we could cut out copies of the sketches. We also obtained some clip-art pictures to represent places such as the farmer's field, the village, the SCPCL office, etc. In a second (smaller) workshop we used the personae to create user stories in the form of cartoon strips or storyboards. Figure 2 shows one of the storyboards created.

³ Self Help Group

⁴ The name Munna was derived from the popular Hindi film series involving the always helpful gangster Munna Bhai.

² Names and locations have been changed to preserve anonymity

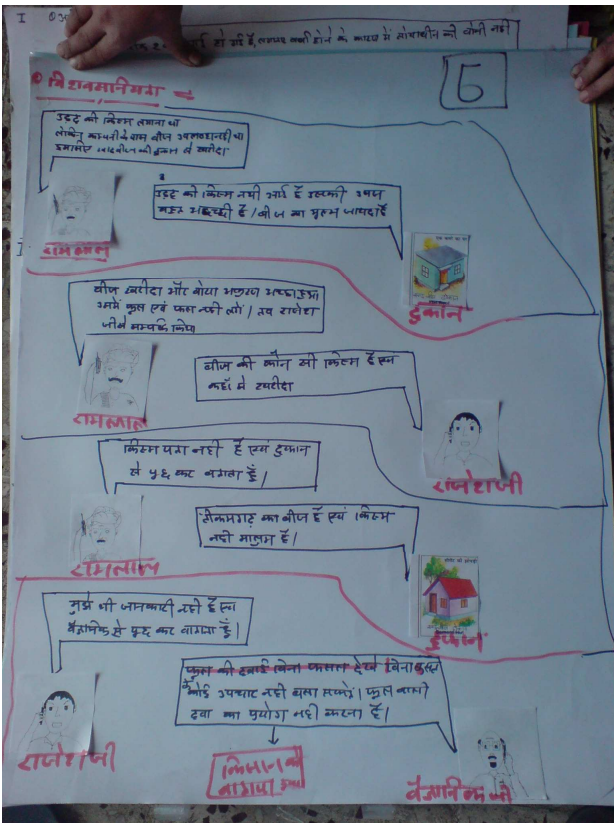


Figure 2: one of the storyboards

Having identified some stories, it was important to establish an understanding of the relative priorities associated with different pieces of functionality. To facilitate these discussions, we used the participatory rural appraisal method of ‘chapatti diagrams’. In this technique, circles of coloured paper (analogous to chapattis), are cut out for each item under consideration. The relative sizes of the chapattis represent the order of priority attached to different ideas, and thus facilitate discussion of relative priority. Figure 3 illustrates the technique in action.

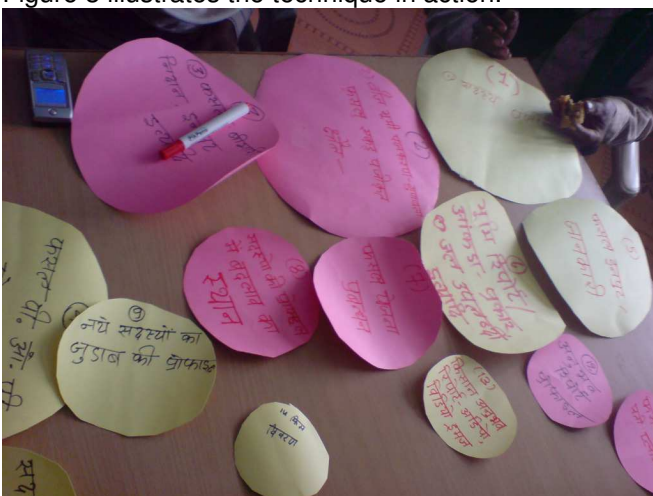


Figure 3: A chapatti diagram discussion

As the project progressed from writing to selecting stories for implementation, we noticed a difficulty with

the role of ‘customer’. A key aim in agile methods is to identify those units of functionality that deliver the maximum value for the minimum cost (Beck, 2005, p. 44ff). However, because the budget was held by the research project, the ‘customers’ were not spending their own money, and so found it difficult to decide which features were most important for themselves. Also, the researcher who was facilitating the discussions began to suspect that the participants were expecting that ultimately all of the functionality would be developed, rather than recognizing that items that they prioritized would be developed, and those that were not prioritized would actually never be built. An added complication was that the funding for the research project allocated a sum for software development, but also had allocations for travel, equipment, and other research activities. Consequently, the precise budget for software had not been completely fixed within the project. A key step was to demarcate a precise budget and share it with the co-operative. With this information, they could make better informed choices, and modified some of their priorities.

Our experience demonstrates how effort is required to enable participants to fulfil the role of customer and user. Similar experiences were reported by early researchers in participatory IT design (e.g. Bødker et al., 1987). Ehn & Kyng (1987) use the term ‘pre-qualification’, to refer to the process of developing the skills needed to participate effectively in software design. Heeks (1999) discusses the problem of ‘resource-deficit’, where a project claims to be participatory, but does not ensure that participants have the capabilities to participate effectively. In this project, we constantly worked to develop the understanding of participants about software designing, but we recognise this as an ongoing learning process requiring regular attention.

Shifting the role of project manager

Communication between developers and the users is a vital factor in software design, so it is important to find a person with the good social skills and empathy to interact with users. At the start of the project, the main project management role for the software development team was held by the managing director of the software company. As the software design progressed, we discovered that this was not satisfactory because of competing demands on the managing director’s time. For this reason, it was necessary to find another staff member to act as the key contact between the facilitator in Sironj and the software designers in AP. Unfortunately, the first person to take over this role resigned shortly after software design began. For the third meeting between the software team and the co-operative, a team travelled from AP including one female member of staff who seemed to have the necessary skills, was able to develop good empathy with the users, and was popular with them. The team discussed her as a potential candidate for project

manager. Unfortunately, she was unwilling to take the role because it involved frequent travel to Sironj an area where communications and mobile phone coverage are poor, and her parents were not satisfied that it was safe for their daughter. Fortunately, another member of staff was found who has taken the role and, with support and capacity building input from the researcher, has performed it well. This incident illustrates how cultural factors and staff mobility, constrain the allocation of critical roles.

Evolving the role of sponsor(s)

Another evolution has been the role of executive sponsor. At the start, the role was located with the principal investigator in the UK, who were paying for the software. This is different from the usual situation in agile software design where the sponsor is a senior person in the users' organisation, which for us would be SCPCL. A third important organisation in our project was the NGO, Pradan, with a head office in Delhi, a local office in Sironj, and a state office in another district of MP.

When the project began field work, one person (Manju) was both team leader for Pradan in Sironj, and a director and Chief Executive in SCPCL. Manju seemed a natural choice as executive sponsor. However, this might not be ideal, because it risked SCPCL members becoming dependent on Manju's leadership. During the project, Manju was offered a senior position in another organisation and left Pradan (and SCPCL). When we discovered that Manju was planning to leave, we had to identify other sponsors within Pradan and/or SCPCL, and to obtain formal commitments to sustain the work after completion of the research.

We also realised, that there would be organisational costs associated with the technology. In particular, SCPCL would need an 'Agricultural Communication Specialist' – someone combining knowledge of agriculture with strong IT skills. With Manju leaving SCPCL the agricultural advisor Ramu was promoted to Chief Executive Officer. This meant that he could not also adopt the role of Agricultural Communication Specialist. However, there was no money in the project to pay this new member of staff. It was necessary to persuade Pradan and SCPCL to commit to covering these costs. The process of negotiating agreements helped in gradually establishing stronger local ownership. In the new configuration, a senior Pradan manager acted as executive sponsor within Pradan, and Ramu, acted as sponsor in SCPCL.

We recognized that Ramu would have a major influence on project outcomes, but the project had to compete with other demands on Ramu's time and attention. It was therefore important for the project to lend support and advice to Ramu to help him develop skills and confidence in his new job as CEO, and to provide encouragement and positive feedback for his engagement with the project.

Our experience suggests that future projects that are initiated from outside of their beneficiary organisations,

need to plan how the role of sponsor will progress over the project lifetime, and monitor progress. It is important that the role of sponsor should be vigorously taken up by people within the beneficiary organisation. A formal memorandum of understanding should be developed with the beneficiary organisation to ensure that there is commitment to the project from the start. Without such active commitment then the intervention cannot be sustainable.

The role of the interaction designer

The role of interaction designers is not defined in the primary descriptions of most agile methodologies. XP simply notes that roles of consultant and boss are external to the development team (see Dubinsky & Hazzan, 2004). As noted above, our project did not have resources available to employ a trained interaction designer to support our work. This expertise could only be offered by the principal investigator, located in the UK, acting as an external consultant. In our project, we did discover a need for interaction design input to support initial designs, and for occasional expert input to resolve impasses, and to open up design possibilities.

One incident arose in designing the interface to capture the multimedia messages on the mobile phone. Our initial ideas were influenced by the Storybank software (Jones et al. 2008) which provides an easy to use interface to create a 'story' composed of six photographs and an audio track. To help with design, the facilitator and some software designers visited the Storybank project to examine their solution. The Storybank software is implemented in Java for one particular phone but was not portable to other similar phones. Our project was implementing using the Python interpreter for Symbian 60 phones (to simplify prototyping and increase portability). However, when the first version of the software was completed, the users found our interface far too complex. The interface required the user to save each photograph under a different file name, then create an audio track, then compose the message by retrieving all the different files. However, when the project manager and developers were challenged to simplify the interface, they could only suggest using a single picture with a single audio file. Discussions with the users indicated that this was not satisfactory. When asked why it was not possible to replicate the Storybank interface, the replies became technical, including issues of multi-threading. The facilitator was unable to respond at this technical level, and so had to retreat from the discussion.

The problem remained unresolved until the principal investigator, entered into the discussions. Before the discussion, he studied the Application Programming Interface (API) for the Python interpreter. Thus, when the discussion began, he could challenge the programmers' assertions about what was or was not possible. As a result of these challenges, and follow up queries, we discovered that multi-threading had arisen

because the programmers assumed that the pictures and audio had to be synchronised (as with Storybank). This would indeed require a multi-threading application on the phone, which would be difficult to implement using the Python interpreter. By removing this constraint, it became possible to implement a simpler interface that was suitable for the intended users.

The motivational role of facilitator

We discovered that the role of the facilitator was much more than just moderating a dialogue between users and software developers. ICT for development implies that the users look for and adopt new practices around ICT. Therefore, the facilitator must motivate users to invest time and energy to explore potential new practices. Such investments have opportunity costs, and there is a constant need to understand the (actual and perceived) costs and benefits for different stakeholders, and respond accordingly.

To fulfil this role, the facilitator needs to be trusted by all participants. Anybody external to the community is always perceived with suspicion and the facilitator was the key referral point for the project. Confidence is essential so that participants trust that the facilitator to protect their interests. Without high levels of trust, it is unlikely that the participants will commit strongly to the organizational and practical changes that new technology demands. On the other hand, the facilitator must avoid creating dependency, rather than empowering the users to act for themselves.

Building trust is a gradual and extended process. For example after initial rapport building, the facilitator participated in weekly and monthly meetings of the co-op and facilitated discussions so that ordinary members could articulate their issues in front of the Directors and to reach to agreeable solutions. The facilitator participated in a *Kisan Mela* (Farmer Fair) organised by the cooperative in a remote area to increase membership. On the personal front the facilitator extended help to individuals. On one occasion some SCPCL members were stuck in Bhopal (state capital) because of a *hartal* (transport strike). The facilitator used his local connections to arrange to get them home (a three hour journey). From this position of trust, the facilitator was then able to motivate and mobilise different actors at different times to progress the work.

One incident that highlighted the importance of this close engagement arose when Ramu took over as CEO of SCPCL. In his new position, there were many competing demands on his time, and the facilitator became concerned that the Rural e-Services technology was slipping down his list of priorities.

One hypothesis was that Ramu was finding that learning his new role was very demanding, and because the facilitator was established in the organization and in the community, he might be able to lead the intervention himself. If Ramu could simply delegate the project completely to the facilitator, with minimal input from himself, then this would free him to

deal with other issues. However, because the project was a short term intervention to provide new software systems, it could not provide that sort of support. It was critical that the co-op adopted these responsibilities internally.

Another hypothesis was that Ramu might think that credit for success would be associated with the Manju not Ramu. Shifting Ramu's perceptions to highlight the potential benefits both to the co-op and to himself was important in capturing Ramu's attention.

Being able to identify and diagnose this situation, to respond to it, and to be listened to when suggesting ways forward, was only possible because of the deep engagement and trust between the facilitator, the co-operative staff and members over an extended period.

The problematic role of product owner

A key role in agile methodologies is that of product owner. The product owner is an authorised person who acts as the final decision maker about priorities and choices in each software cycle. In the rural e-services project this role was fulfilled by the researcher working in Sironj, on the basis that the researcher had a sufficient understanding of software to guide the community in making informed choices, whilst not working for the software delivery organisation and therefore being able to represent the community's interests in negotiations.

However, the choices and priorities had to be guided by wider objectives of community and organisational development. Thus the project to create new software was only one part of a larger project. In fact, the Rural e-Services project could be understood as three related activities: a research activity, a development activity, and a software design activity. Each of these activities can be described (from some perspectives) as 'the project', but each has different aims, objectives and success criteria. For the research project, the aims are concerned with developing knowledge about designing ICT in a development context that then be shared with others through publications and other outputs. For the development activity, the aims are about effective change in the social and economic situation of people in Sironj, through capacity building in SCPCL. For the software design activity, the aims are concerned with timely delivery of working software meeting customers' needs. Each of these projects requires direction and leadership. Coordinating and aligning these objectives is a constant challenge.

Leading the development project is not simply managing the tasks of designing, but also involves managing the expectations, wishes, demands, frustrations, conflicts, motives of the users and other stakeholders. In the early phases of our project, when we conceived of the software development organisation providing the primary source of project leadership, this arrangement was found to be ineffective. The software project manager could not interact with key stakeholders sufficiently regularly to manage and negotiate expectations. Only the locally

based researcher, with regular day to day engagement with SCPCL, Pradan and the farmers was in a position to do this. Also, because the interests of users and customers are not the same as those of software developers, the researcher had to take a role as an advocate to promote users' interests in design. Additionally, the researcher had to review the skills, capabilities and performance of the software development team, and work to build their capacity to interact and communicate effectively with the users. Finally, to ensure successful deployment of the technology, the researcher had to work with the directors, the farmers, the agricultural advisor and the Munnas to devise ways of operating and using the technology that would deliver real benefits, and to encourage the new CEO to adopt a positive attitude in the role project sponsor.

To describe this complex role as simply 'product owner' or 'visionary user' or even as 'facilitator', understates the task. A more realistic title for the role is that of Project Co-ordinator for the development activity. The Development Project Co-ordinator occupies the central role in orchestrating, encouraging and enabling the diverse activities of the users, customer and sponsor. The Development Project Co-ordinator needs to focus on the wider social development objectives of the project and integrate the software development within this broader intervention.

Around this role, it may be possible to identify separate roles of 'product owner', and 'facilitator' who would then work with the development project co-ordinator, however in our project, these roles were closely integrated in one person.

DISCUSSION

Defined roles in software design methodologies should be understood as a useful advisory perspective, rather than a prescriptive framework. As Beck (2005) notes, defined roles should not prevent individual team members from making constructive contributions whenever they are able to do so. Nodder & Neilsen (2009) argue that roles and techniques in agile methods should be understood as resources to help decision making, but that the working practices that are adopted in particular organisations should be based on experience in the particular context, rather than religiously following definitions set out 'by the book'. It is also the case that demarcations between roles and the mapping between roles and individual people in a project will must be sensitive to context, experience and skills. However, our experience with the roles described for agile software development suggests that these existing frameworks are insufficient to guide effective participatory interaction design in social and economic development projects.

Agile development methods assume that customers are powerful and informed actors who are able to promote their own interests in negotiating each stage of software design. The concern of agile software

design organisations is to "... satisfy the customer through early and continuous delivery of valuable software." (www.agilemanifesto.org). Customer satisfaction is critical because customers and users can take their business elsewhere. The model of rapid feedback cycles and regular "planning games" is provided to allow customers and users to constantly reflect on their priorities and review their choices, and so maximise the value they obtain in the process. However, in designing ICT for development, where funds for technology come from outside the user organisation and the organisation begins with limited technology experience, these assumptions are not valid. The customers are not powerful and informed actors at the start of the project. The project must undertake specific efforts to enable them to realise their power, understand their options, and build long term capability. Maunder et al. (2007) reach similar conclusions, advocating an active focus on developing users' capabilities as an essential component of their User Centred Design for Development (UCD4Dev) approach.

Agile methods focus on delivery of satisfactory software systems, not on the wider issues of socio-technical change in the customer's organization. For the proponents of agile methods, these concerns are issues of management within the customer organisation and beyond the scope of software development methodologies. Consequently, the role descriptions set out are primarily concerned with the internal organization of the software development team and software delivery organisation. There is little guidance about role divisions within the customer's organisation, which will vary in different domains. From the software developers' perspective, the main points about the customer's organisation are that there must be users who can provide clarification about issues that affect the software design, and there must be a clear design authority to make decisions about priorities.

Participatory design techniques typically provide a role for a facilitator to work with the users and other stakeholders to inform and guide design discussions, but this role is not explicitly defined in terms of leading change management within the host organisation. However, if the goals of a project are seen in terms of wider social development (as must be the case in interaction design for international development), then it is necessary to identify roles within the project specifically concerned with ensuring that these objectives are achieved. This is a different role from facilitating design.

Our original thinking in the project was that the combination of participatory methods, together with the spread of roles drawn from agile methods would be sufficient for this project. However, we discovered that the role frameworks offered by agile methods do not provide adequate guidance regarding the roles within the 'customer' organisations to manage the organisational changes that are associated with the technology interventions. In commercial settings, customers may hire a consultant to assist them in

negotiating with technology providers and managing organisational change. Such roles are beyond the scope of the role frameworks offered for agile software methodologies. If we wish to describe and codify approaches by which interaction design interventions can contribute to social and economic development, we will need to develop role frameworks that extend beyond the existing scope of agile methods.

In this project, we identified one critical role as that of Development Project Co-ordinator. To introduce this role, it is important to consider the role should relate to the roles of facilitator, sponsor and product owner; how this role can support the development of the user and customer roles; how the role should interact with the software design team and with interaction designers.

Based on our findings, we conclude that future projects in interaction design for international development should pay particular attention to how this particular role is conceived, performed and supported.

ACKNOWLEDGEMENTS

We thank all of the participants in SCPCL, Pradan and the Rural e-Services project without whom this research would not be possible. This work was financed by EPSRC as part of the Bridging the Global Digital Divide initiative.

REFERENCES

P. Abrahamsson, O. Salo, J. Ronkainen & J Warsta, *Agile Software Development Methods: Review and Analysis*, Technical Research Centre of Finland, VTT Publications 478, Finland, 2002.

R Baskerville & MD Myers (Eds.) Special Issue on Action Research in Information Systems. *MIS Quarterly* 28(3), 2004.

R Baskerville & AT Wood-Harper. Diversity in information systems action research methods. *European Journal of Information Systems* 7(1) 90 – 107, 1998.

K. Beck. *Extreme Programming Explained*, Addison Wesley, 2000.

K. Beck (with C Andres). *Extreme Programming Explained* (2nd Ed.), Addison Wesley, 2005.

S. Bødker, P. Ehn, J. Kammersgaard, M Kyng & Y Sundblad. A UTOPIAN Experience: On Design of Powerful Computer-Based Tools for Skilled Graphic Workers. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and Democracy--a Scandinavian Challenge* (pp. 251 - 278). Avebury: Aldershot.

J. Braa, E. Monteiro & S Sahay, Networks of Action: Sustainable Health Information Systems Across Developing Countries. *MIS Quarterly* 28(3), 337 – 362, 2004.

S. Chamberlain, H. Sharp & N. Maiden, *Towards a Framework for Integrating Agile Development and User Centred Design, Extreme Programming and Agile Processes in Software Engineering*, LNCS 4044, Springer, Berlin, 2006.

P. Checkland, & S. Holwell, *Action Research, Its Nature and Validity. Systems Practice and Action Research*, 11(1) 9 – 21, 1997.

A. Cooper, *The Inmates are Running the Asylum*, SAMS, 1999.

A Dearden, X Fu, P Matthews, S Gupta, S Wills. Rural e-services: Participatory co-design of Sustainable Software and Business Systems in Rural Co-operatives. EPSRC Project proposal, A available from: <http://www.odi.org.uk/projects/documents/1068-project-details.pdf> 2009.

A. Dearden & H Rizvi. Participatory Design and Participatory Development: A comparative review. In *PDC'08: Proceedings of the Participatory Design Conference*, Indiana University Press, Bloomington: IN., pp 81 -91, 2008.

Y. Dubinsky & O. Hazzan. Roles in Agile Software Development Teams. In *Extreme Programming and Agile Processes in Software Engineering*. Springer, LNCS 3092, pp. 157-165, 2004.

Ehn, P., & Kyng, M. The collective resource approach to system design. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), *Computers and Democracy--a Scandinavian Challenge* (pp. 17-57). Avebury: Aldershot, 1987

Ferreira, J., Noble, J & Biddle, R. Agile Development Iterations and UI Design. In *Proceedings of Agile 2007*, IEEE, Washington, DC., 2007.

Heeks R. *The Tyranny of Participation in Information Systems: Learning from Development Projects*. Working paper no. 4 in Development Informatics series, University of Manchester, ISBN: 1 9025 1820 9, 1999.

Jones, M., Harwood, W., Bainbridge, D., Buchanan, G., Frohlich, D., Rachovides, D., Frank, M., and Lalmas, M. "Narrowcast yourself": designing for community storytelling in a rural Indian context. In *Procs. of DIS '08*. ACM, New York, NY, 369-378, 2008.

J. C. Lee & D. S. McCrickard, *Towards Extreme(ly) Usable Software: Exploring Tensions between Usability and Agile Software Development*. In, *Proceedings of Agile 2007*, IEEE Washington, DC, 2007.

A. Maunder, G. Marsden, D. Gruijters, & E. Blake. Designing interactive systems for the developing world - Reflections on user centered design. In *Proceedings of ICTD 2007*, pp. 321-328.

C. Nodder & J Neilsen. Agile Usability, best practice for user experience on agile development projects. Neilsen Norman group, Fremont CA. Available from: <http://www.nngroup.com/reports/agile>, 2009.

J. Preece, H. Sharp & Y Rogers, *Interaction Design* (2nd Ed), Addison Wesley, 2007.