Chapter 8

ICT for development: extending computing design concepts

Ulrike Rivett, Gary Marsden and Edwin Blake

Information Communication Technologies (ICTs) such as cellphones have become a part of our daily lives. The technology has rapidly been adopted throughout the world, and this is particularly true in Africa. ICTs provide opportunities for development but the expectations they tend to raise, of improved services and a general modernisation, are often not realised.

The failure rate of ICT systems in the developing world is astonishingly high (Heeks, 2002). Initiatives such as telecentres (centres in rural villages with computers and internet access) have been shown to fail both in the software engineering sense (they are unsustainable and need repair), and in terms of human–computer interaction (no one uses the computers as they are seen as unnecessary) (Benjamin, 2001). Disciplines such as Software Engineering and Human-Computer Interaction have been successful in creating ICT systems for the developed world. In Africa, however, the picture is very different.

Together with colleagues we have developed ICTs for under-resourced and rural communities in South Africa and other developing countries; a field that has become known as Information and Communications Technology for Development (ICT4D). Our focus has been on using devices such as cellphones to develop context-appropriate and easily accessible tools that can support communities and governments in advancing their developmental goals.

In this chapter we provide some examples of our successes and failures in designing ICT4D systems. In assessing these examples, we question basic concepts used in computing such as efficiency and effectiveness, as well as scalability and efficacy. In addition, we explain why we seek to include the voices of actors other than engineers in our design processes. Echoing the findings of Thesen and Cooper in this volume, we show that questions of assessment invariably involve issues of power and location, and that some voices, generally of the most marginal actors in a process, can easily be overlooked.

Some systems are reported as successful when they work in a single location, even if their deployment does not move beyond a single community (Sørenson et al., 2008). However, many existing system design methodologies do not respond satisfactorily when applied to specific development contexts. A major reason for this is that the assumptions on which such systems are based do not fit all contexts, and the outcomes they generate end up being unsustainable or inappropriate. South African society, for example, is characterised by the diversity of its population groups, languages, cultures and religions. Yet, most system methodologies assume that societies are homogenous, and adopt a one-size-fits-all approach, that is designed to be scaled throughout a society. In a heterogeneous context, such as South Africa, system methodologies have to respond to the diversity that exists rather than expecting society to adapt to the system by becoming more homogenous.

Our quest in this chapter is to explore what happens when we build a system, by first identifying a need, and then ensuring that the system addresses the need appropriately through a blend of methods from many disciplines. The end point (and goal) we aim to achieve is feasibility and sustainability. We probe current methods of identifying starting points that can lead to our goals of feasibility and sustainability within a local context. We then explore end points, critiquing existing measures of achievement, and suggesting new ways of understanding success in ICT projects in a social-development context. Our approach involves exploring the contexts in which ICT systems are envisaged as solutions, and how the

disciplines of human–computer interaction and software engineering respond to this approach to system development. We also explore how the existing methods can be improved and made more effective through the use of methodologies such as action research. We conclude that contemporary understandings of success need to be challenged and changed, and engineers and software designers in developing contexts need to recruit users as co-designers, and work with them to agree on the problem to be solved, the means of solving it, and contribute to determining measures of success.

In a water quality project in Cambodia, Mozambique, South Africa and Vietnam, which we discuss in further detail below, the goal was to ensure the delivery of safe drinking water to rural communities. Approximately 1.8 million people die annually from water-borne diseases, with the developing countries of Asia and Southern Africa having the highest mortality rates. Ensuring the provision of safe drinking water at a national level requires not only an understanding of the chemistry and microbiology of water, but also of the workings of institutionalized water systems.

An assumption was made that drinking water quality and its testing are well understood globally, given the influence of the Millennium Development Goals and the efforts of the World Health Organization. However, we identified a gap between the real and perceived needs for water-quality monitoring, as well as the effectiveness of water testing across the different countries. While ICT engineers identified what they considered to be an essential tool in improving the water-quality testing process, this did not necessarily accord with the organizational structures in each of the countries that managed drinking water delivery. While the quality of drinking water might be globally defined, social, political and institutional relationships define how a tool is used, even in contexts in which international standards seem to be accepted and well understood.

What should be built? Modernisation as a desired path?

When analysing existing ICT systems, much of the discussion is positioned within the discourse of development-as-modernisation (Gurumurthy and Singh, 2009; Moodley, 2009). This discourse assumes that modernisation is a universally desired path for economic and technological growth, and that such growth will maximise the achievements of advanced capitalism (located primarily in the industrial North) by transferring these to the periphery. While such discourse is highly contested within the terrain of development studies, it is virtually taken for granted in the ICT field. Most projects follow a top-down, ‘functionalist’ model (Hirschheim and Klein, 1989), seeing ICTs as instrumental and essentially value-free. The ultimate goal of systems development is seen as creating efficient and effective technical systems, exemplifying what Avegerou (2002) calls the ‘techno-economic rationality of western modernity’. Wilson (1997), drawing on the work of Habermas, sees positivist ICT research as adding a scientific gloss to the implementation of technical systems that are presented as being politically neutral.

When investigating ICT systems that have failed, it is apparent that the initial suggestion to develop and implement a system often comes from a technocratic source, such as a research institution, a donor agency or a government department. The failure of such ICT systems is often attributed to background of the user community, and the question of the appropriateness of the ICT system as a solution is rarely tackled. Failure is analysed by assessing aspects such as staffing skills, infrastructure limitations and financial constraints, which speak to the shortcomings of the user community rather than of the system itself (Heeks, 2002). An analysis of the relative power and intentions of the stakeholders

suggesting a system, versus the stakeholders who are expected to use the system, rarely forms part of analysing design, development, success or failure.

The case study that follows provides an example of an ICT solution that considered the influence of various stakeholders as part of the system design. This enabled the design team to evaluate the likely relevance of the tool in relation to the various stakeholders.

The Water Quality Reporter: who are the stakeholders?

The iCOMMS (Information for Community Oriented Municipal Services) team, which forms part of the Department of Civil Engineering at the University of Cape Town, developed a cellphone application called the Water Quality Reporter between 2008 and 2011. The development of the application formed part of the Aquatest project, which was funded by the Bill and Melinda Gates Foundation. The purpose of the application was to allow water-supply managers in remote rural villages to provide information on the water quality to government officials. The cellphone application was implemented and tested in the rural areas of South Africa, Cambodia, Vietnam and Mozambique.

As part of the design process, we analysed stakeholders by developing an importance/influence matrix. The first matrix as shown in Figure 1 focused on those stakeholders that are routinely considered when investigating the challenge of water quality monitoring in a country. (Loudon and Rivett, 2010).

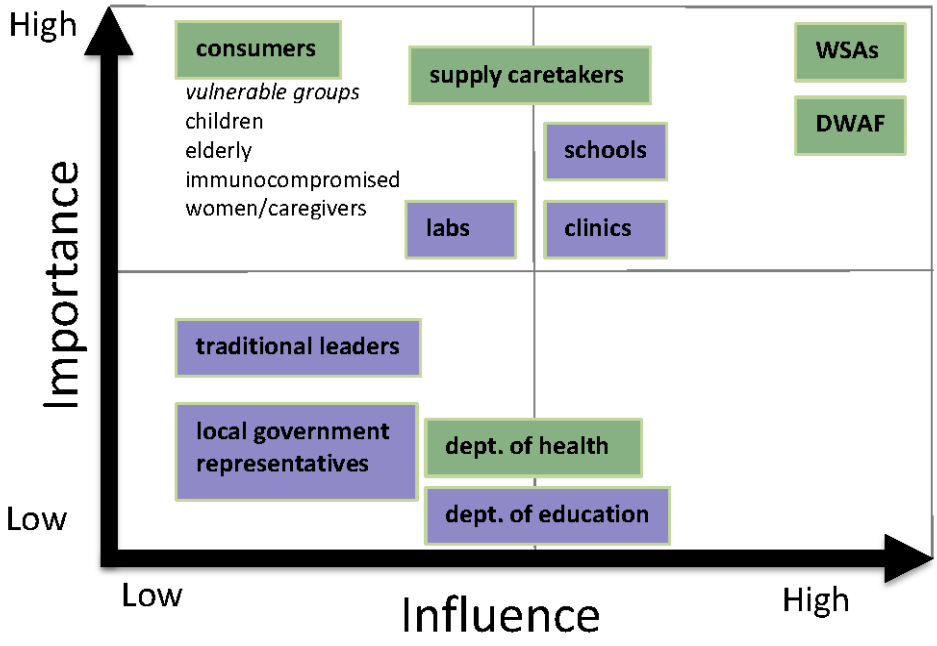

Figure 1 Initial assessment of stakeholders.
Note: DWAF stands for the Department of Water Affairs, WSA stands for Water Service Authority

However, when reviewing the matrix, based on the engagement with the various groups, it became apparent that our own group, namely researchers, had been ignored as a high-importance, high-influence stakeholder. Equally, the funder and the overall project consortium had a high influence in the design of the system, and they too had been excluded from the matrix. A revised stakeholder matrix was then drawn up, which represented the situation in a far more appropriate way (see Figure 2).

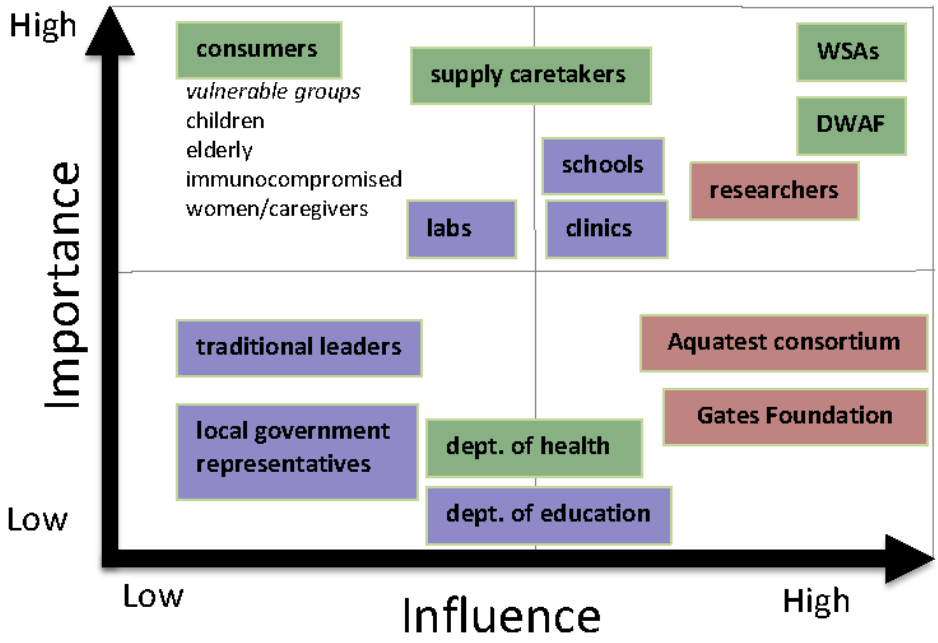

Figure 2 Revised assessment of stakeholders

The more realistic representation of the influence of the various stakeholders gave us a greater understanding of the impact of each group's power and intentions on the design of the system. The need for information flow in the water-quality sector is relatively well defined, based on the roles and responsibilities prescribed by legislation, as well as a fairly widely held understanding and acceptance of what constitutes best practise. Challenges may still arise in the implementation of legislation, given different local arrangements, but there is general agreement on the information flow. Among the researchers, the project consortium and the funders, roles and spheres of influences are continuously negotiated, and the ICT-systems we developed invariably reflected this. For example, we developed a technically challenging mobile-image processing system (Loudon et al., 2009) that was of great interest to the funders, the project team and ourselves, but it did not satisfy any of the particular needs expressed by other South African stakeholders such as the Water Services Authorities, the Department of Water Affairs or local communities.

Obtaining a clear analysis of stakeholders at the outset of a system design, makes it possible to understand the many influences a system has to satisfy. It also allows a more critical understanding of real and perceived needs. Whereas a system design might have to respond to certain requirements for project–political reasons, a 'Southern theory' approach (Connell, 2007) allows us to assess the impact that such a design, and each of the stakeholders, has at various points in the process. At the outset, the identification of an ICT system as a solution to a particular societal need requires that we critically consider who suggested the solution, and how much influence they have on the overall design. Similarly, when a particular system has been identified as a solution, the interests of those involved in its design and development have to be investigated.

Users as co-designers

Within the disciplines of Human–Computer Interaction and Interaction Design, much has been published on how to create appropriate systems for users in the developing world. However, these approaches tend to assume that ICT users, and the tasks that ICTs are employed for, are largely universal. In terms of this model, context is seen as irrelevant. For

example, the approach adopted in *International User Interfaces* (Del Gado and Nielsen, 1996) took a very mechanical approach to creating systems for different cultures. Tasks were assumed to be identical, even if the language through which tasks were mediated might vary. It was thus seen as acceptable to simply translate elements of a user interface from a source language into a target language: so the 'home' key in an English interface becomes a '*maison*' key in a French interface.

The notion of interface translations have been shown to be severely lacking when applied to the diversity of needs and cultures of around 80 per cent of the planet's population that now own a cellular handset (ITU Telecom World, 2011). Users often struggle to understand unfamiliar icons or the concepts they represent. For example, the work of Gitau et al. (2010) shows us that the main reason South African participants in her research could not access the internet from their cellphones was simply because the icons used were inappropriate and misleading.

In an effort to overcome the shortcomings of these superficial notions of culture, many ICT designers in the developing world started to use ethnographic design-led methods (created for well-resourced, primarily urban Northern contexts) to try to uncover the needs of ICT users in developing contexts. However, these methods have been found to be based on similarly mechanical notions of culture (Kimaro and Titlestad, 2008).

It is essential, therefore, that the designers develop some common ground with the end users before they engage in any design. This interaction-design approach attempts to focus broadly on: (i) observing users in a naturalistic setting to ascertain their needs; and (ii) designing new systems and building prototypes, preferably with input from such users.

The idea behind observing users in their own contexts is predicated on the notion that some systems are already in place in their environments. Whether or not these systems are ICT related, the assumption is that the new system is being created to improve on an existing one. In many situations in the developing world, there are few systems in place on which one can base a design: for example, the HPI Research School at the University of Cape Town in 2011 created a system allowing users to create CVs using their cellphones, and then submit the CVs to prospective employers. Prior to this system being built, users had no access to systems that would allow them to create a CV or find jobs online (see www.ummeli.com).

In relation to designing new systems with input from end users, it is very difficult to engage users in co-design when they may have no idea of how ICTs work, or even of the basic separation between hardware and software. Co-design methods also advocate, for example, coming together to sketch ideas on paper, yet in some contexts participants refused to participate, claiming that the process was wasteful of paper (Bidwell et al., 2010). In other words, a co-design approach can also fail if the users' context is not carefully factored in. In short, existing human–computer interaction methodologies cannot show us where to start; cultural models are too abstract and irrelevant, whilst ethnographically inspired design methods are predicated on assumptions that do not hold.

The imperative for users to be co-designers is also manifest in software engineering (that is, methods developed to produce quality computer software). What constitutes good quality changes from 'fit for purpose' to an increasing understanding that clients or users find it very difficult to say exactly what the purpose of a product should be at the outset.

The first attempts at requirements analysis in the early 1980s (Liskov and Berzins, 1986) attempted to elicit very detailed requirements from users by means of careful and painstaking analysis. When this proved unsatisfactory as a way of dealing with complex design problems, cyclical or iterative methodologies were adopted, whereby intermediate systems were developed and presented to users in partial fulfilment of their requirements.

Based on feedback from users, systems are then refined further until the systems and the desired outcomes converge. Not only does this approach address users' inadequate understandings of how computer systems might meet their requirements, it also allows unexpected technical problems to be addressed in a flexible fashion.

These iterative methodologies are collectively known as 'agile' software engineering. Central to such methods is communication between users and engineers. For example, a method, known as 'Extreme Programming', emphasises discovering common metaphors that allow users and engineers to understand how systems work. However, when such methods are used to build ICT4D systems, 'users' and 'developers' may not share a common basis for communication. Sometimes, the very means of communication has to be discovered by both parties. Thus the foundations for agile engineering development may not exist in developing-country contexts, where a common language and understanding still has to be created. This process is frequently assisted by an *intermediary*, a *human access point* or a *champion*, who serves as a 'translator' between users and developers. In our experience, such an agent has to be firmly based within the user community, and must be able to identify the possibilities ICTs offer without raising expectations on either side.

The case study that follows highlights the challenges that communication can pose to users and developers. The missing link of local-context information resulted in a system failing, and then being re-contextualised by the community so that the system is now useful to them, even though it does not fulfil the purposes it was designed for.

Teleconsultation in rural South Africa: who are the co-designers?

The University of Cape Town's Department of Computer Science designed a teleconsultation system (Tucker and Blake, 2008) to enable nurses at a rural health clinic in South Africa's Eastern Cape province to consult doctors at the local hospital, and thus provide better health care to local people. A feature of rural areas in South Africa is that people are not concentrated in small towns or villages, but are spread out in scattered settlements throughout the countryside. Rural hospitals support satellite clinics situated in these settlements.

While designing the system, we successfully solved several technical challenges involved in building a reliable, long-range WiFi network, that allows users to make calls via the internet. We built a mixed synchronous (real-time) and asynchronous (store-and-forward) communication system. Solar-power and battery backups overcame the problem of frequent power outages. We installed relay stations in secure places (such as the headman's house in the settlement where the clinic was located). The asynchronous feature helped over-burdened doctors and nurses who could rarely talk in real-time. We iteratively improved the user interface from a basic design on a laptop to an eventual system that worked via mobile phones.

From an agile design point of view, the system was effective and reliable. The network remained live without interruption for many months at a time. The system was usable and designed to fit into the work patterns and availability of the users. Nevertheless, the system was never used on an ongoing basis. As a teleconsultation system it cannot count as a success.

It has since become clear that there were a few issues that we did not anticipate. For example, despite stating that they used the system, logs revealed that the nurses seldom did. One possible explanation for this is that, in local culture, one does not criticise, and it is seen as important to remain agreeable and connected with the researchers. Eventually, we also discovered that the nurses believed that the system allowed doctors to check on their attendance at the clinic, so they distrusted it, seeing it as a source of managerial surveillance.

We had observed that the nurses were frequently absent when they should have been on duty. One reasons for this is that the local health department seems to have a policy of appointing nurses to posts far away from their homes. For these nurses, getting home on weekends involved an expensive and time-consuming trip of several hours on bad roads. Thus Fridays and Mondays could be lost to travel time.

Much later, however, the system was put to use for an altogether different purpose. Since it allowed some people in the community to gain access to the internet, the doctors added a satellite connection for their own use, and our WiFi system gave internet access to other people along the network, notably the headman's son. Other nodes were subsequently added to the network to provide more coverage for communication services to other settlements and a local community-supported backpackers operation.

We suspect that had we linked the clinics to one another from the start, so that the nurses could talk to each other, the attitude to technology may have been different. However our mindset at the outset was to set up a doctor-nurse consultation system and so linking clinics for chat between nurses was not initially an apparent need. It turned out subsequently that an earlier (and abandoned because no longer supported) citizen-band radio from the previous decade had apparently enjoyed more acceptance possibly because it allowed inter-clinic discussions.

This example illustrates how technology can sometimes not fit in with local communities even if, up to a point, participative and agile approaches have been used to design the system. Rapidly changing local circumstances, combined with difficult-to-detect (and even unconscious) subterranean social circumstances, require a constant interrogation of processes, as well as a willingness to rethink both the starting point of a design and its final outcome.

Technology and 'good ideas' may not be appropriate for local communities, but this does not necessarily mean that local communities won't appropriate a 'failed' system for their own ends. Even the notion of what defines a 'failed' system has to be interrogated further.

How do we evaluate ICT4D? The limitations

One of the challenges of assessing the success or failure of ICT4D systems is the approach adopted to evaluating success. For example, users often define the success of a system by how it responds to their needs, noting simply that 'it works'. This refers mainly to a technical 'working'; that is, the system does not crash and it responds – at least for some stakeholders – to the identified need. As noted earlier, designers and funders assess and evaluate systems in terms of efficiency and effectiveness as well as efficacy and scalability. However, we would argue that these evaluation methods have limitations that can result in ICT4D design remaining stagnant.

'It works'

In engineering terms, software is considered a success when it meets performance metrics related to efficiency or robustness. So, for example, telephone switching software might be considered successful if it takes less than a second to establish a call, and drops fewer than 0.001 per cent of calls. Typically these success metrics are set at the start of a development process and software engineers consider the software complete when those metrics have been met. In the context of ICT4D, however, there are two critical problems with this approach.

The first relates to the notion that success metrics can be defined at the start of a design process. In the technology-saturated, developed world, it is often apparent what technology is missing, and therefore, the steps involved in resolving any deficiencies can easily be identified. In the developing world, however, the introduction of technology radically alters other structures and practices, making it impossible to foresee all that may be required of a technology before it is introduced.

The second problem relates to spurious success. Much software engineering is built on the premise that ICT problems exist outside of, and apart from, any development process. A problem is given to engineers who go off to create the best solution that they can, while assuming that 'development problems' will be solved by 'other' people (such as anthropologists, economists or sociologists). Therefore success metrics tend to be defined in terms of the technology and success is spuriously claimed when these metrics are achieved, regardless of whether users actually benefit or not. Coming back to the example of telephone-switch software, it is possible to build a piece of software that is efficient in terms of dropped calls and connection times, but does not fulfil any useful purpose such as allowing people to chat remotely. For ICT4D to be effective, on the other hand, it must:

- be measured in terms of the impact it makes on users (not just in terms of the performance of the technology);
- acknowledge that problems change over time and that a 'working' solution may only be one step towards a better understanding of the real problem.

Efficiency and effectiveness

The discipline of Human–Computer Interaction overcomes some of the limitations found in other software-engineering approaches, as it centres success criteria on the end user of a system. The internationally agreed standard for usability requires designers to create systems that are 'effective, efficient and satisfactory' for a user to use. While this definition has the user at its core, it does not require that any steps be taken to ensure that the correct problem is being solved.

This can be illustrated with reference to the teleconsultation case study described earlier. Our initial prototype was PC based, requiring nurses to undergo six months training to become comfortable with an unfamiliar device (the PC), the operating system (Windows), and our software. Seeking to improve efficiency, effectiveness and satisfaction, we worked with the nurses to develop a mobile phone application with the same functionality as the desktop system. This required only two weeks of training, and met all the usability criteria. However, even this efficient, effective and satisfying system was never used by the nurses, as we simply did not understand well enough their perceptions of their power in relation to the doctors at the hospital. Essentially, no system built along similar lines would have succeeded.

The failure of usability and much human–computer interaction comes from the focus on the individual, with the intention of optimising the individual's performance. If ICTs are to be developed to serve communities, then success criteria need to be determined by, and for, the community as a whole, and not just by and for individuals from within that community.

Scalability and efficacy

In the context of system design, scalability refers to the notion of being able to use the same system multiple times, either in different settings or for a variety of users. A good example of scalability is Microsoft Office, which is used globally and can be described as a highly scalable product.

Scalability is possible when a system is developed against a highly homogenous background, or when users are able to change their workflow to match that of the developed system. If a system provides enough incentive (for example, when word processors allow us all to write neater and faster), the system is more likely to be easily scalable than when the incentive is less clear. Scalability requires the user to be able to respond to various system requirements, including being able to afford the necessary hardware or software.

Notions of homogeneity and users adapting to system requirements are profoundly in conflict with the needs of communities in under-resourced settings. South Africa has a highly heterogeneous population, to which our eleven official languages bear testimony. In addition, rural communities respond differently to daily challenges based on the resources available to them. A system that responds to the needs of one community may not respond to the needs of another. The notion of creating a single system that can be used by many is not wrong when one can expect the users to adapt. For example, a finance software package that is used throughout the world may require organisations to change their workflows, and result in the software creating a level of homogeneity across communities of users. Yet, in a developing-world context, the notion of 'one system for all', tends to arrogantly assume that the solutions for seemingly similar challenges in different contexts are the same.

Efficacy is another challenging notion when considering system success in the context of ICT4D. Efficacy refers to the notion of how many people 'like' a system, and the degree to which it fulfils the intended outcome. In the context of the Water Quality Report described above, it was possible to assess how much information was flowing between water-supply caretakers and municipalities. However, it was not possible to assess whether the information had an impact on improving attitudes towards water quality within the community, or even whether community health improved as more information was collected. This has been a key problem in a number of system developments. The tracking of the Millennium Development Goals is a good example. The success of Millennium Goal 7C – to ensure environmental sustainability by halving the proportion of the population without sustainable access to safe drinking water and basic sanitation by 2015 – is assessed by tracking how many people have been given access to safe water. However, we don't know if these people still have access to safe water – all we know is that, at one point in time, they had access, but the pump may have since broken. Existing methods of determining efficacy are, to a great extent, self-referential, and the degree to which they assist in developing systems that truly improve design is doubtful.

The beginning of the end and the end of the beginning: methodologies for building ICT4D systems

As we have shown in this chapter, from the starting point of identifying a need for a system to the end point of feasibility and sustainability, ICT4D systems are defined and constrained by software engineering and design. The purpose of design and engineering is to modify reality in ways that suit users and transform their lives for the better in some way. This is what we mean by our extended understanding of 'fit for purpose'. The engineering endeavour then focuses on uncovering a basic common understanding between all participants, and the design methodology used is intended to facilitate this. While consciously aiming at transformation via our engineered interventions we critically question which purposes at the beginning and end of the process are best suited as goals for designs.

From an engineering or design perspective, we believe that far more emphasis has to be put on mutual learning and knowledge discovery. We never lose sight of the primary aim of design, which is to facilitate improvement of some aspect of the lives of users. From a

research perspective, on the other hand, we are intent on intervening in the lives of users while continually increasing our understanding of the situation, our discipline and ourselves.

Our solution to the issue of goal setting lies in involving the community where our intervention is to become active. We regard the community as partners in design, and we call this aspect of the method 'community-based co-design'. By having the community as co-designers, we start to address the issues identified above, namely:

- Understanding the community's needs: having community members as active partners in the design process means that we stand a greater chance of understanding and responding to their needs;
- Avoiding paternalistic notions of development: by engaging the community directly, we do not need to rely on external, top-down, notions of development.

The question, of course, is how best to identify and engage potential users in this new methodology. Several options are possible. We make use of three, depending on the nature of the various situations we work in.

The first is to communicate with communities via an intermediary; that is, a person or a group (such as an NGO) who have the trust and respect of a community. In this situation, communication shifts from being a dyadic exchange between co-designers towards a triadic interpreted design of a system artefact. This has important consequences, and leads to a mediated design process.

The second is to build partnerships with community members (with or without the help of intermediaries) that result in deep mutual understanding. In such situations, the co-design conversation is easier because concepts and values are shared. This is a very long-term process, and can require decades rather than years. This approach raises important ethical issues of reciprocity between engineers and community members.

The third involves splitting the design process between building tools to support artefact production, and the production of the system artefacts themselves. This means that, rather than always creating a finished solution, we may create tools that enable people to create their own solutions. The tools are also created with the community in a joint design process. The essential idea is that engineers limit their role to procuring tools, and training others in the use of those tools; the community then takes over the tools to produce their own systems. This co-design conversation, which mostly involves an intermediary, has the benefit of being more short term, and is aimed at discovering needs and training requirements.

The epistemology of effectiveness

Engineers understand 'fit for purpose' in terms of the extent to which a system meets the stated needs of users. Once we accept that identifying such needs is very difficult, and that once we include users as co-designers, then discovering a practical purpose for which an artefact can be designed is problematic. It is also possible that the academic needs of researchers to know and understand will conflict with the needs of the community.

Our 'solution' to this is to conflate *practice* with *knowing*. Creating knowledge is inextricably intertwined with effective action. Knowledge that does not lead to effective action is not really knowledge, and the failure to create effective systems is equivalent to a failure of understanding. This is the position of pragmatist epistemology – an epistemology that is compatible with action research (Oquist, 1978).

Conclusion

We have argued that mediated design processes, or user-needs analyses that include communities as co-designers, result in a more holistic approach to software design and engineering. The shift represents a move from a closed system of expertise, with the engineer as the expert and other participants as subjects, to open collaboration and co-ownership of the design process. This challenges the traditional role of the designer profoundly. Instead of the designer setting the agenda, deciding on the methodologies and owning the outcomes, the emphasis moves to a shared learning approach. This fundamental shift results in the designer giving up control, and becoming a facilitator.

References

Avgerou, Chrisanthi (2000) 'Recognising alternative rationalities in the deployment of information systems' *Electronic Journal of Information Systems in Developing Countries*, 3 (7), p. 1–15.

Benjamin, Peter (2001) 'Community development and democratisation through information technology: Building the new South Africa' in Richard Heeks ed., *Re-inventing Government in the Information Age*. London: Routledge.

Bidwell, Nicola J, Thomas Reitmaier, Gary Marsden and Susan Hansen (2010) 'Designing with mobile digital storytelling in rural Africa' in *Proceedings of the 28th International Conference on Human Factors in Computing Systems* (CHI '10). New York: ACM.

Connell, Raewyn (2007) *Southern Theory: The Global Dynamics of Knowledge in Social Science*. Sydney: Allen & Unwin.

Del Gado, Elisa *and Jakob* Nielsen (1996) *International User Interfaces. New York: John Wiley & Sons.*

Gurumurthy, Anita and Parminda Jeet Singh (2009) 'ICTD: Is it a new species of development?' Bangalore: IT For Change, http://www.comminit.com/ict-4-development/node/306946 Accessed online 26 November 2012.

Heeks, Richard (2002) 'Information systems and developing countries: Failure, success and local improvisation' *The Information Society*, 18 (2), p. 101–112.

Hirschheim, Rudy and Heinz K Klein (1989) 'Four paradigms of information systems development' *Communications of the ACM*, 32 (10), p. 1199–1216.

ITU Telecom World, *The World in 2011: ICT Facts and Figures*, http://www.itu.int/ITU-D/ict/facts/2011/material/ICTFactsFigures2011.pdf Accessed online 28 November 2012.

Kimaro, Honest and Titlestad, Ola (2008) 'Challenges of user participation in the design of a computer based system: the possibility of participatory customisation in low income countries' *Health Informatics in Developing Countries*, 2(1), p. 1–9.

Liskov, Barbara Huberman and Valdis Berzins, V (1986) 'An Appraisal of Program Specifications' in Gehani Narain and Andrew McGettrick, eds, *Software Specification Techniques*. New York: Addison Wesley.

Loudon, Melissa and Ulrike Rivett (2010) 'Methods and conceptual tools for success in local government information systems', a paper presented at the IDIA Conference, Cape Town , 3-5 November 2010. www.developmentinformatics.org/conferences/2010

Loudon, Melissa, Tahmina Ajmal, Ulrike Rivett, Dirk de Jager, Robert Bain, Robert Matthews and Stephen Gundry (2009) 'A "human-in-the-loop" mobile image recognition application for rapid scanning of water quality test results', a paper presented at the First International Workshop on Expressive Interactions for Sustainability and Empowerment (EISE 2009), 29–30 October, London, http://ewic.bcs.org/content/ConWebDoc/33637. Accessed online 6 March 2013

Moodley, Sagren (2005) 'The promise of e-development? A critical assessment of the state of ICT-for-poverty-reduction discourse in South Africa' *Perspectives on Global Development and Technology*, 4 (1), p. 1–26.

Oquist, Paul (1978) 'The epistemology of action research' *Acta Sociologica*, 21 (2), p. 143–163.

Sørensen Tove, Ulrike Rivett and Jill Fortuin (2008) 'Review of e-health systems for HIV/AIDS and antiretroviral treatment management in South Africa' *Journal of Telemedicine and Telecare*, 14 (1), p. 37–41.

Gitau, Shikoh, Gary Marsden, and Jonathan Donner (2010) 'After access: Challenges facing mobile-only internet users in the developing world', *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI 2010)*. New York: ACM.

Tucker, William and Edwin Blake (2008) 'The role of outcome mapping in developing a rural telemedicine system' in Paul Cunningham and Miriam Cunningham eds, *IST-Africa 2008: Conference Proceedings*. Windhoek: International Information Management Corporation.

Wilson, FA (1997) 'The truth is out there: The search for emancipatory principles in information systems design' *Information Technology & People*, 10 (3), p. 187–204.