Software Engineering in Developing Communities

Edwin Blake
University of Cape Town
Dept Computer Science
Rondebosch, 7701
South Africa
+27 21 650 3661

edwin@acm.org

ABSTRACT

This paper reflects on our work in deriving targeted methodologies to develop IT applications and content in a developing world environment. This paper argues that a common thread over more than a decade of experience in building Information and Communication Technology systems has been a community centred approach. We relate this to the African philosophy of ubuntu. These approaches are wrapped into an iterative Action Research paradigm to include the communities of users directly.

Categories and Subject Descriptors

D.2.m [**Software Engineering**]: Community Based Software Methods

General Terms

Management, Design, Human Factors.

Keywords

Action Research, Computer Science, Developing world, Ethics, Software Engineering, Ubuntu.

1. DESIGNING APPLICATIONS THAT SUPPORT DEVELOPMENT

This paper is concerned with identifying the essential underlying themes that have governed our approach to providing Information and Communications Technology (ICT) applications for a developing country such as South Africa. Our intention is to trace the threads of the ideas as they developed. We can start in 1996 with two broad ideas: express all requirements within the context of a community and use an iterative action research based method to develop solutions.

The contribution of this paper is to show one way in which deeply held beliefs about the importance of community can be operationalized into a Software Engineering method. From a (South) African perspective the challenge is to re-vision western methods to accord more closely with local values. The difficulty is that these multi-facetted local values were originally expressed in stories [15]. For Africans then the challenge is how do we turn the insight, captured in the term *ubuntu*, that our humanity is

intimately interconnected with, and dependent on, the community of the living and the dead[1] into something that helps us develop software? In this paper we document the processes that lead us to such a method.

The contribution of the paper should have wider appeal however since the Software Engineering community is beginning to embrace similar values: witness the success of the workshop series "Cooperative and Human Aspects of Software Engineering". It is only in retrospect that we realized the importance of ubuntu in guiding our research: it was too much in the air at first for us to name the source. This process of extracting normative values for Software Engineering from implicitly held views of a community might also be of wider interest.

We have named our method "Socially Aware Software Engineering" [5]. As our point of departure we wanted to align technical and socio-cultural factors. This socially aware software engineering method is part of our research in Information and Communications Technology for Development (ICT4D). This method is a based on action research and was used to guide cyclical interventions with target communities to solve community problems with ICT.

For example, one feature we had to capture in our Software Engineering framework was to explain why users might not accept a technically sound ICT system and develop a socially aware approach that accounted for such facts. We found that we have to enable, and even encourage, unintended uses of software artefacts that empowered users to appropriate information technologies on their own.

Blake and Tucker [5] described initial thoughts on an approach that combined HCI methods, participatory design and software prototyping under the umbrella of action research. As mentioned elsewhere [13], methods like participatory design are problematic until participants acquire sufficient ICT literacy. We use the prototypes as a tool to facilitate communication between participants and the development team. Users cannot help design, but they can actively participate.

---

[1] Ubuntu is Zulu and means "humanness" (humaneness?); the same word and concept in other South African languages: "Botho" in Sesotho, "Vhuthu" in Venda, and "Umntu" in Xhosa. Elsewhere in Africa: "Ujamaa" Kiswahili (Tanzania), "Unhu" Shona (Zimbabwe), "Utu" Swahili (Kenya), "Abantu" Ugandan. Archbishop Desmond Tutu said it means: "My humanity is caught up, is inextricably bound up in yours."

2. ORIGINS OF COMMUNITY BASED INFORMATION TECHNOLOGY

In outlining a South African position on Information Technology for the G7 ISAD Conference in 1996 [16], we first enunciated a divergence from the then popular views of the information revolution: Information Community; different from the "Information Superhighway" (USA) or "Information Society" (Europe). We identified four key issues:

Information Community (IC) Perspective: This focuses on the implications for social groups as well as individuals. The Information Revolution should benefit society as a whole.

The Role of the Information Community in Promoting Development: The development needs of each country, as set out in South Africa by the RDP, should be the focus for the IC.

Ensuring equity in the Information Community: The IC must be an instrument of emancipation and empowerment, appropriate to the environment and needs of each country.

The role of the state in the Information community: The state has a role to play in ensuring universal access and supporting establishing the economic environment for the IC.

Our document is clear on the social aspects of systems development:

A key issue for developing countries is to have the effective power to determine their own IT futures. That power only derives from controlling the process of IT development. How IT is applied, and whose interests it serves, is central to whether this is an enabling technology, or a sophisticated way of keeping people and communities disempowered. At various stages in the development and use of applications, it is necessary to ensure that the needs of the end users are paramount, and not those of intermediaries. One aim must be to break down the barrier between IT professionals and "ignorant users". All have a stake in IT.

Specifically on applications development:

The field of applications development must be dramatically broadened beyond the traditional mathematical and engineering approach. Social and human aspects must also be taken into account.

The earliest lesson that we learnt therefore was that an approach to ICT in Development has to be community based: a notion of an "Information Community" as opposed to the individualistic approach of the developed world to computing. For example, the very concrete notion that contrasts the provision of personal computers to everyone to a more community-based approach that recognizes providing *access* to ICT resources as the more relevant objective.

3. USE ACTION RESEARCH

We believe that one should use Action Research for ICT Development when one has to act but still also needs to learn. Action research seems to be a methodology that attracts forceful adherents and detractors. This is surprising since action research applies an engineering paradigm for dealing with complex design situations in order to achieve democratic social reconstruction. It seems the ideal way of addressing situations where designers do not initially understand local issues and culture and where, at the same time, the local communities cannot appreciate the potential of ICT to address their development needs.

The cyclical approach to action and reflective learning has been around for a long time[2]. It was advocated in its various forms in the post Second World War development of the UK, USA and Japan. We believe that developing countries currently have similar urgent (re-)construction needs; it is not surprising therefore that a radical return to similar frameworks for dealing with the situation might be useful again.

Action Research grew out of the work of two groups after the Second World War: Kurt Lewin (a refugee from Nazi Germany) who founded the Research Centre for Group Dynamics at MIT and researchers at the Tavistock Institute of Human Relations affiliated to London University. The war had encouraged cross-disciplinary work on tackling complex problems with an emphasis on attaining results and learning from the experience. These groups set out to apply these lessons to broader social issues [1][12][11].

We have built up some experience in the use of Action Research in ICT projects. We started with the CyberTracker project in 1996 [3] and we are currently busy with communication projects involving the deprived Deaf communities in Cape Town and Health workers in a remote part of the Eastern Cape Province of South Africa [4]. The overall aim of our various ICT development projects is to investigate ways for developing useful systems for socio-economic development in deprived areas. This is essentially a design problem. (Design, of itself, sits somewhat awkwardly within traditional computer science, see [2] for further discussion).

We use action research not exactly as a methodology for research but more of as a mindset, an attitude. If one looks at the literature, Action Research comes in many flavours. All of which seem to be different takes on how to use the method in practice in different situations. In our case we wish to place more emphasis on the Action (or Engineering) aspect than is often done while retaining the key importance of learning and reflecting.

One issue that arises when one does this is that there must be recognized metrics to evaluate the effectiveness of an intervention: we have been looking at Monitoring and Evaluation (M&E) Tools for this purpose The advantage is that such tools have focussed for a long time on development issues and are sophisticated instruments for evaluating impact, sustainability and so forth. We have used both the Real Access/Real Impact criteria of bridges.org and the Outcome Mapping method developed by the IDRC [17]. They are also easy to incorporate into the evaluation phase of an action research cycle.

The problem with using M&E Tools for design is that they do not really give the ICT designer assistance such as, for example, design patterns. More subtly, from the point of view of an ICT designer, M&E tools tend to treat ICT in a static fashion and not as something that is easily mutable and adaptable. This means planning incremental adaptation is more difficult.

---

[2] David Grier recently gave a light hearted overview of the Shewhart cycle: iterate over "Plan; Do; Check; Act (Analyze)" [9]. Shewhart proposed the cycle in the early 1930s; it was popularized by Deming in the 1950s

3.1 Ethics of Action Research

In showing how to incorporate implicit community values into software engineering we first want to discuss the values and ethical issues that arose out of Action Research.

In a publication by Lewin, one of the originators of Action Research, he argued that action research should take as its inspiration a "blueprint … taken from research on physical engineering" [12]. Thus he was arguing for the insights of a design-orientated science to be applied to social issues. We believe that things have come full circle and we can now re-integrate the insight of Action Research into Software Engineering. It is this social dimension that sets Action Research apart from an "Iterative and Agile Software Engineering as usual" approach that some advocate in ICT4D situations.

A key feature of Action Research has been a desire to act ethically as a researcher. Ethics can only be born out of honesty about *ourselves* and the community in which we operate. I would like to relate three insights that have been most difficult for us to accept.

First: The long-term interest of a researcher is not sustainability of the action outcome but well regarded published results. This has to be made clear to all participants at the start of a project. This does not mean that researchers are not interested in sustainability, but in the long run their careers, and in the case of students their degrees, do not depend on sustainability. It is the community as a whole, and perhaps the researchers as part of the community, that has the interest in sustainability. Only once this is honestly admitted can joint strategies with the community be developed to help ensure the long-term support of the system if it is successful.

Second: The community might be very ambivalent to change since it upsets cherished notions and comfortable dysfunction. As Jaques [11] points out:

> *"ambivalent attitudes on the part of the community to the very process of solving its own problems. On the one hand, there is the desire for improvement, for resolution of tension, for the development of new techniques, and, on the other hand, there is the fear that treatment is likely to be more painful than the problem. Exposing the complexity of problems, with the inevitability of creating new roles for which the individuals concerned do not yet feel adequate, excites the desire to hang on to old modes of behaviour. Though the old modes are troublesome, until they are relinquished one can avoid facing up squarely to the fact that perhaps not all is as well as it might be."*

Third: The community reluctance to accept the "beneficial" solutions proposed by the software developer might have to lead to the developers being forced to give up their cherished notions of progress. After all it is the democratic right of a society to filter new products through their cultural and social requirements. Thus the right or wrong of the new application depends on the situated creation of meaning by the community [10][8].

3.2 Engineering or Research?

One can characterize the traditional purpose of Software Engineering as an attempt to reduce and manage uncertainty in software design. One can do this by analysing the problem thoroughly beforehand so as to reduce the possibility of unwanted surprises. The proponents of agile software engineering have pointed out the futility of doing this in complex and unpredictable situations. They advocate a reduction of uncertainty via iterative methods that explore the solution space actively through prototypes.

Our method adds to such an exploratory approach the possibility that, as engineers, we might never be able to remove uncertainty in functionality and thus we leave it up to the users to appropriate our artefacts for their own, unanticipated, uses as equal partners in the enterprise of Action and Research. The goal is to achieve an alignment of multiple perspectives in order to produce artefacts which are valuable to all stakeholders.

This method adds further demands on the ICT4D software project leader or manager. We have found that developing software in the field required substantial effort in coordinating stakeholders, including participants and their coordinating bodies, for example, the government bureaucracies; NGOs and intermediaries; and software developers, graduate students and full-time programmers. ICT4D projects incur disproportionate costs for human resources, hardware and travel. Someone with both technical and social skills needs to manage such ICT4D projects.

4. CONCLUSION

The African philosophy of *ubuntu* provides the grounds for our approach since it regards one's identity as a human being as causally and even metaphysically dependent on a community. These values support looking for a community based approach to ICT. Such an approach cannot be adopted in an uncritical fashion however since core values of privacy may be threatened by the widespread use of ICT, which a naïve community based set of values might not question[18]. This is an issue to be addressed in our research by reflecting on current practises where we would advocate a functional hybrid of these ethical values (see [6] for a review). Our approach to Software Engineering can thus also be seen a small contribution to the attempts to develop the normative implications of ubuntu [14][7].

We also argue for a community-based approach to Software Engineering and ethics. This is an approach in which researchers and target users become equal members of the same community that is intent on developing knowledge through effective action. Each party admits their own legitimate interests in the joint enterprise. It seems to us that such an approach leads to some kind of Critical Action Research where the legitimate needs of the users for action are combined with the equally legitimate needs of the researchers for research results.

The cyclical nature of action research, where questioning and reflection are tied to intervention, neatly solves the need of users to learn about ICT while the researchers learn about the community within which they are working. Many popular ICT development methods have assumptions, frequently unarticulated, about users' knowledge of information technology artefacts, one such is participatory design. Such approaches have largely failed to meet the challenges of ICT Design for Development.

As a result of our experiences we have come to believe that creating knowledge is inextricably intertwined with effective action. We believe that knowledge that does not lead to effective action is not really knowledge and that a failure to create effective systems is equivalent to a failure of understanding.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] Bargal, D. 2006. Personal and intellectual influences leading to Lewin's paradigm of action research: Towards the 60th anniversary of Lewin's 'Action research and minority problems' (1946), Action Research 4; 367. DOI: 10.1177/1476750306070101

[2] Blake, E.H. 2009. Design Doctorate in Computing: a defence of "doing cool stuff". Technical Report CS09-06-00, Department of Computer Science, University of Cape Town. http://pubs.cs.uct.ac.za/archive/00000598/

[3] Blake, E. H. Steventon, L. Edge, J. and Foster, A. 2001. A field computer for animal trackers. In CHI-SA 2001: 2nd South African Conf on Human-Computer Interaction. CHI-SA and ACM SIGCHI, 2001. www.chi-sa.org.za/CHI-SA2001/cybertracker.pdf

[4] Blake, E. H. and Tucker, W. D. 2006. User interfaces for communication bridges across the digital divide. AI & Society, **20**(2):232–242.

[5] Blake, E.H. and Tucker, W.D. 2006. Socially aware software engineering for the developing world. Proc Inf Soc Technologies, (IST-Africa), Pretoria, South Africa.

[6] Cappuro, R. 2007. Information Ethics for and from Africa. International Review of Information Ethics (IRIE). Issue 7, Paper1. URL: www.i-r-i-e.net/inhalt/007/01-capurro.pdf

[7] Cornell, D. 2005. Exploring Ubuntu — Tentative Reflections, African Human Rights Law Journal, 2, 2005. www.fehe.org/index.php?id=281

[8] Dourish, P. 2001. Where the action is: the foundations of embodied interaction, MIT Press, Cambridge, MA.

[9] Grier, D. 2009. Bad Alignment. IEEE Computer **42** (11) Nov, 8–11.

[10] Harrison, S. Tatar, D. and Sengers, P. 2007. The three paradigms of HCI. Proc. Human Factors in Computing Systems, (CHI '07 alt.chi), San Jose, CA, USA.

[11] Jaques, E. 1947. Social Therapy: Technocracy or Collaboration? Journal of Social Issues; **3** (2), 59–66

[12] Lewin, K. 1946. Research on Minority Problems. Technology Review January 1946: 163-190.

[13] Maunder, A. Marsden, G. Gruijters, D. and Blake, E.H. Designing interactive systems for the developing world — reflections on user-centred design. In Proceedings IEEE/ACM Int Conf on Information and Communication Technologies and Development (ICTD2007), 321-328, Bangalore, India, 2007.

[14] Metz, T. 2007. Toward an African Moral Theory, Journal of Political Philosophy: 15: 3, 2007, 321–341. DOI: 10.1111/j.1467-9760.2007.00280.x

[15] Murove, M.F. (ed.) 2009. African Ethics: An Anthology of Comparative and Applied Ethics. University of KwaZulu-Natal Press, Scottsville, South Africa

[16] NITF (National Information Technology Forum). 1996. The Information Society and the Developing World: A South African Approach. Dept of Arts, Culture, Science and Technology, 1996. South African position paper at the G7 Information Society and Development Conference, Gallagher Estates, Midrand, 13-15 May 1996. No longer available on-line but can be found at (thank you) web.archive.org/web/19990210065934/www.sn.apc.org/nitf/pptoc.htm and www.cs.uct.ac.za/~edwin/MyBib/1996-isad.pdf

[17] Tucker, W. D. and Blake, E. H. 2008. The role of outcome mapping in developing a rural telemedicine system. In Paul Cunningham and Miriam Cunningham, editors, IST-Africa 2008 Conf Proc, Windhoek, Namibia, 2008. IIMC. www.ist-africa.org/Conference2008/outbox/ISTAfrica_Paper_ref_67_doc_2116.pdf

[18] Olinger, H.N. Britz, J.J. and Olivier, M.S. 2007. Western privacy and/or Ubuntu? Some critical comments on the influences in the forthcoming data privacy bill in South Africa. The International Information & Library Review, 3: 1, March 2007, pp 31-43. DOI: 10.1016/j.iilr.2002.08.001