Requirements Engineering at the Margins: Avoiding Technological Hubris through Alternative Approaches

Debra Richardson
Department of Informatics
University of California, Irvine
+1 949-824-7353
djr@ics.uci.edu

Ban Al-Ani
Department of Informatics
University of California, Irvine
+1 714-824-8090
balani@ics.uci.edu

Hadar Ziv
Department of Informatics
University of California, Irvine
+1 949-824-4047
ziv@ics.uci.edu

ABSTRACT
Technological hubris occurs when attempts are made to develop technological solutions for marginalized groups. Despite being impoverished, these groups constitute the bulk of Information and Communication Technology (ICT) users. Developers all too often assume these peripheral, marginalized groups have the same needs as people in the core, developed countries and engineer technologies accordingly. Likewise, software engineers typically use the same approaches to elicit requirements and develop technologies for such groups. Both these tactics run the risk of disregarding the true needs of such users by not taking their environment, social order, or influences of either into account. Our position is that developers must reconsider current, widely adopted requirements engineering approaches when developing ICT for marginalized groups. We advocate embracing alternative techniques from the social sciences, here considering two such techniques, namely cultural probes and storytelling. We explore how these techniques can be adapted for software requirements.

Categories and Subject Descriptors
D.2.1 [Software Engineering]: Requirements/Specifications; K.4.0 [Computers and Society]: General; K.4.2 [Computers and Society]: Social Issues.

General Terms
Design, Human Factors.

Keywords
Marginalized groups, requirements engineering, cultural probes, story-telling.

1. INTRODUCTION
According to the introductory comments for the NSF Workshop, "Creating a Research Agenda in Computing at the Margins":

> *While Information and Communications Technologies (ICTs) have demonstrated their ability to transform some societies, others remain on the computational margins. One approach to addressing the challenge of* ***Computing at the Margins*** *has been to make scientific and policy assessments that argue for increased access to, and infrastructure support for, the same technologies found in more technologically saturated sectors of society. But results of this approach have reported mixed results. Further, the "increased access" approach also closes off research opportunities to examine whether, and if so how, scientific and methodological practices may themselves be contributing to the persistence of the digital divide and the breakdown in technology transfer that it continues to represent.*
>
> *Evidence mounts that what is actually required is scientific and methodological innovation predicated on the assumption that the digital divide represents unique infrastructural, computational, and societal challenges that when accounted for in the methods and theories of Computer Science lead to impact. ... Evidence shows that domains that have typically been outside the scope of Computer Science research (i.e., those that have been at the margins of scientific investigation such as countries in the Global South and underserved demographics in developed nations), provide powerful catalysts for scientific and methodological innovations because of their unique infrastructural, computational, and societal differences.*

Clearly, the same is true for requirements engineering – that is, innovative approaches to requirements engineering are required in the context of ICT development for people at the margins. [1]

Requirements engineering (RE) activities identify and analyze user and business needs, stakeholder goals, product objectives, and so on. Software engineering process models that include human/user-centered concerns explicitly focus on user needs and usability issues. Within the context of engineering for marginalized people, our observation is that even these explicit steps are subject to the implicit assumptions often made by "westerners" – developed people – such as "We know what you need", "We know how to do it right", "We know what will improve your quality of life", and so on. When this attitude comes into play in developing countries and deploying software and information systems for marginalized groups, there is a tangible risk of *technological hubris*. Our position is that future software engineering approaches must change to appropriately address the needs of and develop technologies for marginalized groups.

Today's software and, more generally, information and communication technologies (ICT), play a growing, integral role in supporting people in virtually every country and of virtually every socio-economic status. Yet while ICT development is

FoSER 2010, November 7–8, 2010, Santa Fe, New Mexico, USA.

[1] We use the terms "at the margins" or "marginalized" to include people who reside in less developed or developing countries or non-industrialized nations, and also impoverished, underserved people in developed, industrialized nations.

certainly driven by a global market, it is usually designed and engineered by a select few countries. These are the developed countries and are generally considered "western". Thus, existing requirements engineering approaches usually reflect how needs can be elicited from people in western-developed countries, and ultimately reflect western developers' perceptions of user needs and what constitutes a "better quality of life" [18].

Approaches to requirements engineering (RE) must take into account the target users' context and environment to better facilitate developing ICT that meets a real need of the user group. The need to do so increases as ICT and computing devices become more ubiquitous and the software that drives them permeates more aspects of everyday life. ICT and software are no longer limited to large organizations or to specific countries or regions; rather they have assumed a global reach and penetration and are increasingly developed for a global market. This shift poses a challenge to our typical assumption that software developed for one country or one target user-base can be easily adapted to the needs of others by a simple exchange of character sets, icons, colors and other superficial aspects of the product. We know from the work of Nielsen and others that making software international and global is not at all easy [3].

We use *marginalized groups* to refer to people who reside in less developed or developing countries or non-industrialized nations, and also impoverished, underserved people in developed, industrialized nations. Previous studies have found that while these groups utilize ICT, technologies are rarely developed with them in mind. Our position is that different and possibly new approaches are needed to develop software for marginalized groups that reduce the likelihood of technological hubris. The challenges of developing software for such groups are numerous. Here, we discuss broad types of challenges, aimed at finding approaches that can be adopted to identify, engage and ultimately understand users in their contexts. We recognize that additional challenges exist beyond the ones we discuss here.

In this paper, we suggest that developers consider approaches typically used in the field of social sciences. The purpose of our research is to identify approaches that enable discovery of the true needs of marginalized groups; to investigate the applicability of those approaches to real marginalized groups; to apply relevant approaches to gathering real user needs for real systems; and to develop prototypes that will support these approaches and thereby produce ICT and software that better meet those needs. This is early work – that is, we do not claim to have yet accomplished this research. Instead, this position paper aims to discuss users' contexts and environments leading to investigating alternative approaches that can be better adapted to engineer requirements for ICT use by marginalized groups.

2. APPROACHES AND RESULTS

Developers have aimed to provide ICT services to marginalized groups and adopted different approaches during the process with varying degrees of success. Existing approaches to system development typically highlight the importance of the type of system to be developed, the developers' competence and/or the availability of the user [16]. Most human-centered and user-centered software development lifecycles include steps that explore users' contexts, characteristics and environments, attempting to genuinely identify and address their true needs. They do not, however, include steps to explore and address cultural, social and other key differences between such user characteristics and their own.

These oversights may stem from the reality that a system is often developed with a specific core group in mind - e.g., people who live in a western or westernized society or in a well-known and well-defined environment [18]. However, the bulk of eventual users of a system may not comply with characteristics projected by the developer, and thus their needs often remain unrepresented. Researchers have reported that these marginalized groups are frequent technology users. For example, 92% of homeless young people sampled in Los Angeles and Denver use "technology" weekly, with social networking sites being accessed 3.8 times/week [22]. As well, 1 million people become mobile phone users everyday, and 85% of them live in the developing world [4].

Widespread ICT use by such groups have drawn the attention of researchers who have responded to the growing adoption by engineering technologies to address the needs of hitherto marginalized groups [18]. While we do not question that the resulting software is useable, we do question the software's usefulness. We question its usefulness because proposed "solutions" to perceived problems experienced by these groups do not always reflect an understanding of the conditions within these environments and consequently how ICT can be utilized to improve target user conditions. In the case of most of the marginalized groups, these conditions are difficult to perceive and it may be even harder to understand how these conditions influence ICT use.

For example, impoverished communities in rural areas of India are known to share cellular phones, as they recognize the need to pool resources limited by their finances [8]. In another instance, it was found that limited bandwidth, which led to slow (and thus costly) Internet access by professionals in Kenya, meant that they needed to have a planned and purposeful interaction style that involves offline preparation [23]. Another example is the use of cellular phones to support religious practices [23]. These projects demonstrate that ICTs are useful to marginalized groups and also that utilizing them typically involves many challenges. We suspect that these projects are rarely initiated by asking people in the target population what their needs are; instead, researchers observe the population and propose a solution accordingly. We draw on two studies to illustrate that, even with best intentions in mind, user needs may not be adequately met.

In the first study [13], researchers extended an existing radio program initiated by a local organization to support marginalized farmers of small plots in a rural region of India. The radio program incorporated listener feedback regarding following the advice broadcast through the program, an opportunity to voice new problems, etc. Here, researchers adopted an existing model and extended it by developing a simple interactive voice response (IVR) toolkit for community discussion application to allow farmers to ask and respond to questions, and to browse others' questions and responses on a range of agricultural topics. The application's success can be traced to researchers going to great lengths to find out not only farmers' needs, but also their affordances [13]. They also had the advantage of utilizing an existing practice that already had an established audience and proven need (the radio program). They extended an existing model without changing features that are essential to its success.

This project is undoubtedly a success with increased end user participation; they benefitted from the information provided through the extension. This success indicates that the resulting system met some need; however, there is no indication that these needs were elicited.

In another study, researchers made a concerted effort to understand the target users, in this case children in rural India, by observing them in their environment – yet they still failed to address their true needs to some extent. These researchers chose to develop an educational cellular phone application, focusing on a particular subset of the group, namely children who do not have the opportunity to attend the regular public school because they were forced to earn a living. The project involved providing these children with mobile phones that are pre-loaded with an application to teach them English [8]. While the researchers' efforts were commendable and resulted in a better understanding of mobile usage patterns and context, the project did little to improve the lives of the underrepresented group – the children. Instead, researchers sought to provide what they believed to be a "gateway" to economic advancement in India which does not necessarily hold true for children who cannot attend school regularly because of the more pressing need to eke out a means to survive (the researchers themselves report that about 43% cannot attend school in rural India.) However, the researchers did not seem to investigate what it is that these children *need to learn,* and thus increased the risk of technological hubris.

Technological hubris therefore appears to be implicitly embedded in development approaches as well as developer perceptions of user needs, and this hubris is almost unavoidably embodied in the resulting technology. We therefore ask: what are the alternatives to such approaches and do they reduce the risk of technological hubris? A new field has emerged – *ICTD* (Information and Communication Technologies and Development) is the application of technological solutions to problems in the developing world. We posit that technological hubris occurs frequently in this domain and that improved requirements engineering approaches, including techniques adapted from the social sciences, are needed to avoid this risk.

3. INVESTIGATING ALTERNATIVES

A review of previous ICTD experiences leads us to conclude that requirements engineers need to *see* what marginalized groups *see* and understand their conditions and needs through their words and actions if we are to develop ICT that meet their true needs. This is typically not accomplished by traditional requirements engineering approaches but can only be done by also studying the dynamics of society, culture and the human condition.

We are not the first to suggest that techniques from social sciences be applied in requirements engineering. In their survey of requirements engineering [12], Nuseibeh and Easterbrook already note that "RE draws on the cognitive and social sciences to provide both theoretical grounding and practical techniques for eliciting and modeling requirements", including, among others, cognitive psychology, anthropology, sociology, and linguistics.

In Viller and Sommerville, ethnographic or ethnographically-informed approaches to RE provide deeper understanding of work as it is actually performed by humans [19, 20]. They consider the limitations of existing ethnographic approaches to be the time and effort they require; the length and detail of reported results; cultural obstacles and differences; and differences in skill levels. They discuss concurrent ethnography, where ethnographers are involved in an iterative prototyping lifecycle, concurrently with software engineers. They consider 'quick and dirty' ethnography and evaluative ethnography to still follow the basic model where social scientists carry out ethnographic studies. They then propose that results of ethnographic studies be communicated differently, using methods and notations more comfortable and more familiar to requirements engineers, namely presentation viewpoints, presentation framework, and the Coherence method [19,20].

Our work is similar in that we also wish for requirements engineering to perform cultural and other analyses themselves, supported by ethnographic studies and methods. If Viller and Sommerville consider ethnographically-informed design as "front end" to traditional requirements analysis, then our work can be viewed as "front end to the front end", where ethnographic studies are aimed at learning about true needs and wishes of marginalized groups, without any assumptions or technological hubris that we know "what they need" or even what questions to ask, prior to conducting and evaluating the results of ethnographic studies.

An earlier paper on ethno-methodology by Goguen and Linde [7] surveys both "traditional" RE techniques, including introspection, questionnaires, interviews, focus groups, and protocol analysis, as well as three discourse analysis techniques from the social sciences. We agree with these authors' premises, interpreted in the context of this position paper, namely that

- understanding the social order of marginalized groups is critical to successful elicitation of their needs and requirements;
- requirements analysts cannot assume they know the social order of a marginalized group or that such users even fit into the same categories - such as social class, norm, role, and so on - as themselves;
- making such assumptions can lead to requirements with serious inaccuracies and misunderstandings.

Instead, ethnographic techniques will prove more useful to understanding the true social order of members of a marginalized group, reducing the likelihood of cultural false positives [11], and leading to requirements that more accurately reflect their needs.

Goguen and Linde [7] emphasize that requirements engineers must consider how the target users categorize themselves to organize their interaction. We suggest that this would help discover how this social order is relevant to a needed ICT product. This can be best achieved by engaging the users in the development process through ethnographic techniques, despite the certainty of encountering challenges when attempting to engage marginalized groups. The rewards of this engagement, however, will not only increase user satisfaction [15] but also address the potential to avoid widening the divide between us (developers) and them (marginalized groups). Minimizing this divide becomes even more important when we recognize that the majority of the "them" reside in developing countries and the "us" reside in the developed world.

4. AN OUTLINE OF OUR APPROACH

We propose adopting a combination of ethnographic approaches to gain insights into the contexts, conditions and environments of

marginalized groups, and to elicit their needs and affordances. We suggest employing two such approaches, namely cultural probes and storytelling. While there may be other potentially beneficial approaches, we will focus on these two initially. Both approaches explore and rely on users' perceptions instead of those of the developers. Insights gained from ethnographic approaches will, in turn, provide input into a more 'traditional' software life cycle, starting with requirements engineering.

Cultural probes [2,5] typically consist of multiple media used by people to capture aspects of interest in their everyday life. They could, for instance, consist of one or more artifacts such as cameras, audio recorders, post-it notes, writing paper, etc. that the target population can use to tangibly capture aspects of interest. We propose handing out kits that consist of disposable cameras, questions and post-it notes to members of the marginalized group of concern. Questions ask members to take photographs of specific aspects of their environment and life. For example, what brings them joy, what challenges them, what would they like to accomplish, what do they want to avoid, and so on. Members who use and return the kits will be rewarded in a manner meaningful to them. Using cameras to capture what is significant in their lives does not require any specialized skill or education, as cameras can be used with minimal training. Post-its can be used to label the items that they photograph, thus they can categorize items as "joy", "challenge" and/or "avoid". Labeled photographs will help us *see*, literally, important and relevant aspects of their lives with minimal investment of training time and money. These kits will help us understand aspects of their lives visually rather than through a more traditional method such as words documented through a written questionnaire or verbal interview. Thus using cultural probes can help reach people who may not be capable or available to respond to a questionnaire or articulate their needs.

Each participant will then be asked to tell the story behind his or her pictures. Information collected through the probes can be used to support and complement the elicitation of needs through storytelling, by acting as prompts to help developers evoke users' points of joy, interest and hardship in their environment. Mobile digital storytelling has been used as a means to gain insights into cross-cultural design by HCI researchers [1]. We aim to adopt this approach to better understand the marginalized groups' environment and other important factors in their lives, as described through their story narrative. The verbal narrative and visual representations afforded by the photographs complement each other and collectively provide a more accurate account of the users' needs. This approach will enable us to hear and perhaps understand some of their experiences. The media produced by cultural probes, as well as collected stories, will provide insights into the real needs, concerns, and problems of the marginalized group being studied. Insights will, in turn, serve as input into software engineering – specifically, requirements engineering – activities aimed at developing software and ICT solutions for their problems. The early use of ethnographic approaches will reduce the risk and likelihood of technological hubris in the development of solutions.

Storytelling is an integral part of most cultures and is often used to incorporate complex aspects of everyday life into stories to explain new and innovative experiences [17]. Storytelling has been used in a wide spectrum of systems, ranging form proactive health management [9] to design for rural areas [1]. Storytelling is also a cornerstone in teaching from kindergarten to higher education, and recently storytelling has been used successfully in computer science education for example using Alice, Storytelling Alice and Scratch [21]. Further, stories have long been used in software development, for example in the form of use cases, user stories, interface design scenarios, and requirements scenarios.

We predict that enabling members of marginalized groups to provide verbal descriptions through storytelling will allow them to articulate their needs more readily. The storytelling process is supported and driven to some extent by the photographs from the cultural probes. Both will increase our understanding of their environments and their true dimensions. The mix of storytelling and cultural probes will also help us to more readily identify similarities and differences across target population and help minimize the risk of cultural false positives. This in turn will help identify common needs shared by the majority of target users as well as common characteristics of those users.

Our ultimate goal is a requirements engineering approach that tackles the core challenge of identifying and addressing the true needs of marginalized groups. We intend to conduct an exploratory study to investigate the feasibility of adapting storytelling and cultural probes toward requirements elicitation. We will start by gathering real needs for a real system to be developed, then develop and deploy a working prototype aimed at meeting those needs. This exploration (and others) will lead to guidelines on effective requirements engineering practices that include ethnographic techniques, after which we propose to develop support for requirements engineering in the ICTD domain. However, we posit that such an approach incorporating ethnographic techniques could also improve requirements engineering practices in general.

More generally, we wish to integrate social science insights and practices into software requirements engineering, starting at the college level with undergraduate- and graduate-level training. The Department of Informatics at the University of California, Irvine, has several professors with background in social sciences including, among others, ethnography, anthropology, cognitive psychology, and more. The undergraduate major in Informatics includes courses in requirements analysis methods as well as social-science methods, HCI, CSCW, and so on; we plan to add materials on marginalized groups and requirements at the margins to existing courses.

Our proposed approach borrows social science techniques, which is more common in the HCI and CSCW communities. For instance, Pitula and Dysart-Gale [14] also focus on socio-cultural differences among user groups and the digital divide between the "haves" and "have-nots". They contend that current HCI practices – as applied to requirements engineering – are too shallow and narrow to truly address the needs of marginalized groups. They suggest that HCI practices must be broadened in order to cover cultural issues more deeply, and draw on Communications Theory to that end. These techniques are rarely adopted within software engineering in general and in requirements engineering in particular. In the future, software engineering will inevitably be employed more widely than ever, given the growing deployment of information and communication technologies in developing countries and under-industrialized nations; to succeed we must embrace such cross-disciplinary, integrated approaches.

REFERENCES

[1] Bidwell, N. J., Reitmaier, T., Marsden, G., and Hansen, S. 2010. Designing with mobile digital storytelling in rural Africa. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems* (Atlanta, GA, April 2010). CHI'10. ACM, pp. 1593-1602.

[2] Boettcher, A. 2006. Moving from cultural probes to agent-oriented requirements engineering. In *Proceedings of the 18th Australia Conference on Computer-Human interaction: Design: Activities, Artefacts and Environments* (Sydney, Australia, November 2006). OZCHI'06. ACM, pp. 253-260.

[3] del Galdo, Elisa M. and Nielson J. *International User Interfaces*. John Wiley & Sons, 1996.

[4] Dörflinger, J., Friedland, C., Merz, C., and de Louw, R. 2009. Requirements of a mobile procurement framework for rural South Africa. In *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems* (Nice, France, September 2009). Mobility'09. ACM, pp. 1-4.

[5] Gaver, B., Dunne, T., and Pacenti, E. Cultural Probes. *ACM interactions*, 6, 1 (January/February 1999), pp. 21-19.

[6] Goguen, J.A., "Social issues in requirements engineering," *Proceedings of IEEE International Symposium on Requirements Engineering* (Bonn, Germany, January 1993). RE'93. pp.194-195.

[7] Goguen, J.A.; Linde, C., "Techniques for requirements elicitation," *Proceedings of IEEE International Symposium on Requirements Engineering* (Bonn, Germany, January 1993). RE'93. pp. 152-164.

[8] Kumar, A., Tewari, A., Shroff, G., Chittamuru, D., Kam, M., and Canny, J. 2010. An exploratory study of unsupervised mobile learning in rural India. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems* (Atlanta, GA, April 2010). CHI'10. ACM, pp. 743-752.

[9] Mamykina, L., Miller, A. D., Mynatt, E. D., and Greenblatt, D. 2010. Constructing identities through storytelling in diabetes management. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems* (Atlanta, GA, April 2010). CHI'10. ACM, pp. 1203-1212.

[10] Medhi, I., Gautama, S. N., and Toyama, K. A comparison of mobile money-transfer UIs for non-literate and semi-literate users. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems* (Boston, MA, April 2009). CHI'09. ACM, pp. 1741-1750.

[11] Mrazek, D. and Baldaccini, C. 1997. Avoiding cultural false positives. *ACM interactions* 4, 4 (July 1997), pp. 19-24.

[12] Nuseibeh, B. and Easterbrook, S. 2000. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (Limerick, Ireland, June 04 - 11, 2000). ICSE '00. ACM, New York, NY, pp. 35-46.

[13] Patel, N., Chittamuru, D., Jain, A., Dave, P., and Parikh, T. S. 2010. Avaaj Otalo: a field study of an interactive voice forum for small farmers in rural India. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems* (Atlanta, GA, April 2010). CHI'10. ACM, pp. 733-742.

[14] Pitula, K. and Dysart-Gale, D. Expanding the Boundaries of HCI: A Case Study in Requirements Engineering for ICT4D. *Information technologies and International Development*, Vol. 6, No. 1, Spring 2010, pp. 78-93.

[15] Robinson, W. N., Pawlowski, S. D., and Volkov, V. Requirements interaction management. *ACM Computing Surveys.* 35, 2 (Jun. 2003), pp. 132-190.

[16] Scacchi, W., Process Models in Software Engineering, revised version to appear in J. Marciniak (ed.), *Encyclopedia of Software Engineering, 2nd. Edition,* 2002. Wiley, pp. 993-1005.

[17] Search, P., Digital storytelling for cross-cultural communication in global networking. Chap. 6 in Selected readings on information design: communication, technology, history and education. Eds: Carla Spinillo; Petrônio Bendito; Stephania Padovani - Curitiba: Sociedade Brasileira de Design da informação, 2009.

[18] Soeftestad, Lars T. and Sein, Maung K. (2003), ICT and Devolopment*:* East is East and West is West and the Twain may yet Meet*.* Published in: Krishna, S. and S. Madon, eds., The Digital Challenge: Information Technology in the Development Context, 2003. Aldershot, United Kingdom: Ashgate, pp. 63-82.

[19] Viller, S. and Sommerville, I. 1999. Social analysis in the requirements engineering process: from ethnography to method. 4th International Symposium on Requirements Engineering (RE'99), Limerick, Ireland, June 1999.

[20] Viller, S. and Sommerville, I. 2000. Ethnographically informed analysis for software engineers. Int. J. Human-Computer Studies, 2000, Vol. 53, pp. 169-196.

[21] Winbladh, K. Precise and Usable Requirements through an Interactive Model-based Approach. PhD Dissertation, University of California, Irvine, CA, 2010.

[22] Woelfer, J. P. and Hendry, D. G. 2010. Homeless young people's experiences with information systems: life and work in a community technology center. In *Proceedings of the 28th international Conference on Human Factors in Computing Systems* (Atlanta, GA, April 2010). CHI'10. ACM, pp. 1291-1300.

[23] Wyche, S. P., Aoki, P. M., and Grinter, R. E. 2008. Re-placing faith: reconsidering the secular-religious use divide in the United States and Kenya. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 2008). CHI'08. ACM, pp. 11-20.