A Software Development Methodology for Sustainable ICTD Solutions

Joerg Doerflinger
SAP Next Business and Technology
Karlsruhe, 76131 Germany
joerg.doerflinger@sap.com

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

Tom Gross
Human-Computer Interaction Group
University of Bamberg
Bamberg, 96045 Germany
tom.gross@uni-bamberg.de

CHI 2013 Extended Abstracts, April 27–May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract

Information and Communication Technology continue to be increasingly used in social development and poverty alleviation projects, known as Information and Communication Technology for Development (ICTD) projects. However, most interventions either fail completely as a result of attempting to use inappropriate software development approaches and technology concepts in the different ICTD context or they only execute small scale prototypes without positive long-term social impact. We present a case study on how we combined and adapted, using an iterative action research refinement approach, established interaction design methods into a software development methodology supporting scalable long-term ICTD software projects: the Technical ICTD Methodology (TIM). Our case study is based on the experiences of a series of ICTD projects executed within a major software corporation over a period of more than five years.

Author Keywords

Software Development Methodology; ICTD

ACM Classification Keywords

D.2.2. Software Engineering: Design tools and techniques.

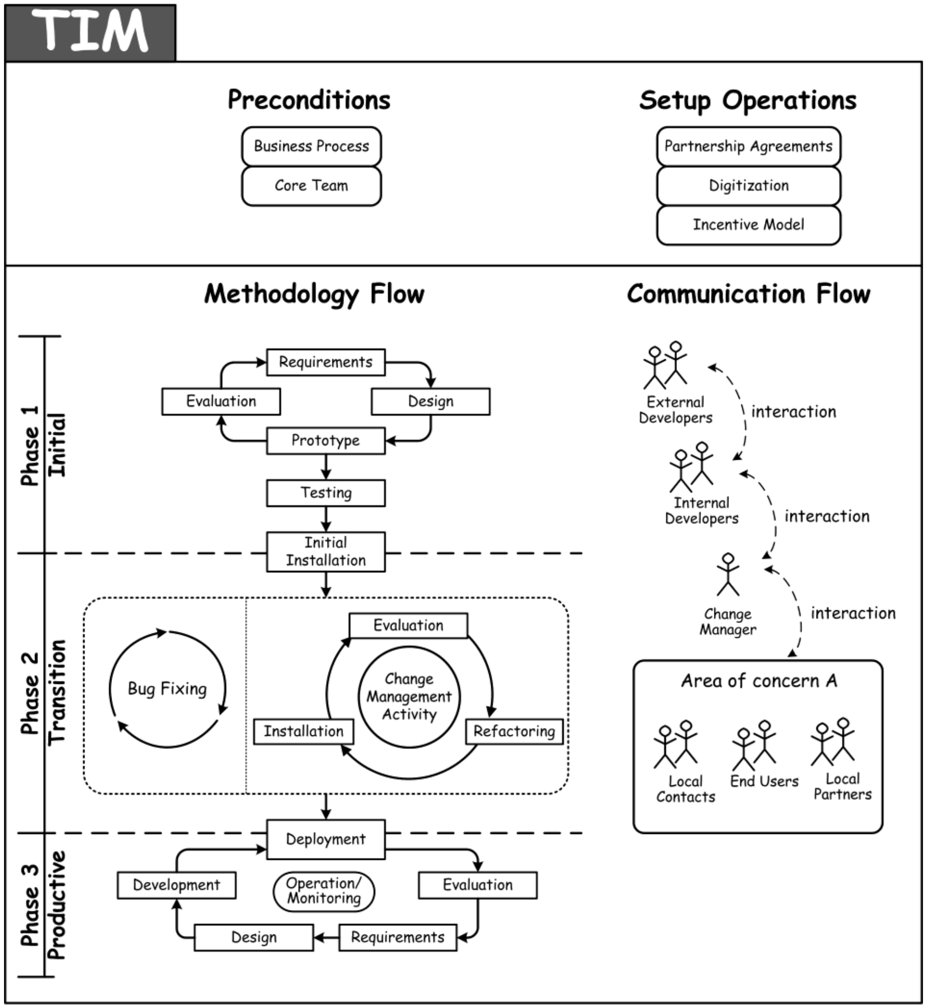

Figure 1: TIM – Technical ICTD Methodology.

Introduction

Technical ICTD, which refers to ICTD software development topics, is missing a replicable and scalable software development methodology tailor-made for the ICTD context. While currently the major focus in technical ICTD centers on technology solutions [6,8,10], less attention has been paid to devising appropriate software development methodologies (though see [1,3,7]). However, to scale beyond small scale prototyping solutions towards sustainable and scalable long-term solutions replicable and scalable methodological approaches are required. We present a software development methodology that has been refined through an iterative action research process within a series of ICTD projects over a period of more than five years - the Technical ICTD Methodology (TIM) (see Figure 1). TIM builds on established interaction design methods and extends them with a number of distinctive features reflecting our learning over these five years, in particular, a multi-phase iterative development approach, a set of role definitions for key actors, and a defined communication flow.

Methodology Refinement Approach

Action research is an established framework for information systems research gaining acceptance in computing research that seeks social relevance [5]. Building on West & Stansfield [11] ‘FMA approach’ we have utilized an iterative action research based process for the continuous refinement of TIM. Following this approach research proceeds as a series of cycles where researchers select an Area of concern (A), then take action in that situation applying the methodology (M), which is guided by a framework of ideas (F). It is important that F and M are made explicit prior to taking action. During and after each cycle, reflection generates findings leading to adaptations to the framework of ideas and the Methodology. Following this approach we started with an initial set of assumptions of how we can benefit from established interaction design methods. However, only along the iterative approach we learned how to adapt and utilize the methods appropriately. We expected to benefit from User-Centered Design (UCD) and Participatory Design (PD) approaches. We started with methods such as rapid/iterative prototyping, personas and scenarios, and on-site field studies. Our framework of ideas included the expectation that those methods would need to be adapted to the cultural and infrastructural context [12] but we would find out how during the iterative refinement. We expected that if we want to build solutions that foster long-term business collaborations with local partners we need a tight collaboration with them from the very beginning [2]. We also assumed that we would benefit from being on-site to build tight personal collaborations with local stakeholders. Along the iterative refinement within three ICTD projects TIM evolved based on the continuous lessons learned about our initial assumptions. TIM distinguishes it from our initial expectations and existing methodologies in four key areas: collaboration setup, defined team roles, communication flow, and three phase process.

Resulting Methodology - TIM

In contrast to initially simple 'Test and Evaluation Agreements' the refined TIM collaboration setup defines collaboration models like 'Tester' that only trials ICT solutions, along 'Co-Innovators' actively participating in the ICT development process, to 'Early Customers' collaborating in the transition phase from research to real business. Beside typical team roles (e.g. software developers) TIM builds on the assumptions of the required local change manager role. While in the early pilot scenarios we had different people present on-site we found that this results in weak trust relationships with local partners, adds extra 'hand-over' effort on return, and introduces different individual interpretations of results. In the refined TIM the change manager role is the primary on-site contact and single design authority throughout a project. She/he interacts with stakeholders, identifies opportunities, and translates them into possible ICT solutions. She/he is external to the local partner and end-user community which contrasts with Sharpey-Schafer's [9] 'champions' who are local stakeholders with a vested interest in driving the project internally, but who are not necessarily able to assess trade-offs between commercial benefits, technological opportunities, and implementation efforts. The TIM communication flow defines the communication hierarchy along project structures. In earlier pilots for example we had direct communication between off-site software developers and local end-users resulting in redundancy and misinterpretation of information caused by missing local context knowledge and a weak trust relationship. In the refined TIM the change manager is the main contact for all local communication. In the initial pilots the software development approach was an undifferentiated iterative software development process. Over the five years it evolved into the defined three phase approach guiding software development along the different needs of prototyping, roll-out, and live operation. Further examples of how the iterative reflections in our projects led to local adaptations of established interaction design methods are: on-site activities executed by local partners or local community members provide more reliable feedback than if led by the change manager alone; a user-trains-user workshop format makes users feel more comfortable and is more effective as a one-on-one change manager-trains-user setup; because of unreliable power, paper prototypes are fall backs to functional prototypes, but an early shift to functional prototypes improves feedback; end-users often try to follow the strongest voice or do not want to express doubts in front of community leaders, so working with segregated groups of users with similar status can be helpful; on-site activities must adhere to local cultural conventions (e.g. praying, welcome the community chief before workshop starts, respecting gender issues).

Conclusion

The positive results of projects applying TIM [4] show TIM to be a repeatable technical ICTD software development methodology, enabling the creation of sustainable, scalable, and reusable ICTD solutions. The positive results strengthen the case for locally adapted interaction design methods in ICTD software development. The utilization of an action research approach for the iterative refinement of TIM helped to cope with the ICTD specific context and to define the methodology based on the iterative learning in real use cases. PD and UCD being the main concepts of TIM enable a tight end-user interaction supporting the creation of appropriate ICT solutions. Extensive field studies resulting in personas and scenarios, establishment of local contacts, and the role of a change manager help to avoid design vs. reality gaps. Rapid prototyping and end-user workshops with local support and early paper prototypes and later functional prototypes ensure the tight integration of end-users in the entire design process. Whilst every software development project is in some ways unique, we believe that other ICTD projects could benefit to draw on elements of the UCD and PD approach of TIM.

Further References

TIM has been designed, refined, and actively used in the following research projects.

ACi: The African Cashew initiative (ACi) project aims to increase the competitiveness of African cashew production in five West African countries. *Video: "Software solutions for developing countries" (http://youtu.be/k32D7UrgkCw) accessed 20th Jan. 2013*

SSN: The Star Shea Network (SSN) is a national federation of shea farmer cooperatives in North-East Ghana. *Video: "Software solutions for developing countries" (http://youtu.be/pWkuaPdXUB0) accessed 20th Jan. 2013*

C@R: Collaboration@Rural (C@R) is a research project aiming to foster socio-economic development of Small, Medium, and Micro Enterprises (SMMEs). *Video: "Merchands go Mobile" (http://youtu.be/bhMyY2gUCQU) accessed 20th Jan. 2013*

References

[1] Blake, E. Software engineering in developing communities. *Proc. 2010 ICSE Workshop - CHASE*, 1–4.

[2] Brewer, E., Demmer, M., Du, B., et al. The case for technology in developing regions. *Computer 38*, 6 (2005), 25–38.

[3] Dearden, A., Rizvi, H., Gupta, S. Roles and responsibilities in agile ICT for development. *Proc. India HCI 2010*, 1–12.

[4] Doerflinger, J., Gross, T. Sustainable ICT in Agricultural Value Chains. *IT Professional*, July/August (2012), 48–55.

[5] Hayes, G.R. The relationship of action research to human-computer interaction. *ACM Transactions on Computer-Human Interaction 18*, 3 (2011), 1–20.

[6] Kumar, R. eChoupals: A Study on the Financial Sustainability of Village Internet Centers in Rural Madhya Pradesh. *ITID 2*, 1 (2004), 45–73.

[7] Maunder, A., Marsden, G., Gruijters, D., Blake, E. Designing interactive systems for the developing world - reflections on user-centred design. *Proc. ICTD 2007*.

[8] Parikh, T.S., Lazowska, E.D. Designing an architecture for delivering mobile information services to the rural developing world. *Proc. WWW 2006*.

[9] Sharpey-Schafer, K. Key Techniques in Building Effective ICT Solutions in the South African HIV/AIDS Sector. *Proc. CCC Workshop*. Berkeley, (2009), 77–78.

[10] Veeraraghavan, R., Yasodhar, N., Toyama, K. Warana Unwired: Replacing PCs with mobile phones in a rural sugarcane cooperative. *Proc. ICTD 2007*, 1–10.

[11] West, D., Stansfield, M.H. Structuring Action and Reflection in Information Systems Action Research Studies Using Checkland ' s FMA Model. *Systemic Practice and Action Research 14*, 3 (2001), 251–281.

[12] Winschiers, H. The Challenges of Participatory Design in an Intercultural Context: Designing for Usability in Namibia. *Proc. PDC '06*, II, 73–76.