Always Beta: Cooperative Design in the Smart Home

Timo Jakobi
University of Siegen
Hölderlinstraße 3
Siegen, 57068 Germany
Timo.Jakobi@uni-siegen.de

Gunnar Stevens
University of Siegen
Hölderlinstraße 3
Siegen, 57068 Germany
Gunar.Stevens@uni-siegen.de

UbiComp '13, September 08 - 12 2013, Zurich, Switzerland

http://dx.doi.org/10.1145/2494091.2497325 2494091.2497325

Abstract
In Software development, the always beta principle is used to successfully develop innovation based on early and continuous user feedback. In this paper we discuss how this principle could be adapted to the special needs of designing for the Smart Home, where we do not just take care of the software, but also release hardware components. In particular, because of the 'materiality' of the Smart Home one could not just make a beta version available on the web, but an essential part of the development process is also to visit the 'beta' users in their home, to build trust, to face the real world issues and provide assistance to make the Smart Home work for them. After presenting our case study, we will then discuss the challenges we faced and how we dealt with them.

Author Keywords
Living Labs, Smart Home, Design Case Study, Software Engineering, Hardware Engineering

ACM Classification Keywords
B.m Design management, D.2.2 Design Tools and Techniques: Evolutionary Prototyping

General Terms
Human Factors, Design

Introduction

In software engineering agile development [1] has become quite popular, where software is always beta and is continuously improved through feedback from users that have had a first-hand experience with the application. In contrast, the design of home automation and Smart Home technology often follows the traditional waterfall development model where users can test a new system only very late in the field-test phase.

In this paper we will show, what adapting "always beta" to the special needs of designing for the Smart Home means, where we not just take care of the software, but also handle hardware components. In particular, because of the 'materiality' of the Smart Home it is not possible to just put a beta version on the web, but an essential part of the development process is to visit the beta-users at home, build trust, face real world issues and provide assistance to make the Smart Home work for individual home infrastructures.

We will illustrate our experiences in a project where we developed a smart energy feedback solution using a Living Lab [2, 3] approach with seven participating households [4, 5]. At the end, we will summarize the lessons we learned and discuss the scope and limitations of our approach.

Smart home Beta

"[A p]roduct is developed in the open, with new features slipstreamed in on a monthly, weekly, or even daily basis. It's no accident that services such as Gmail, [etc. ...] to bear a "Beta" logo for years at a time. Real time monitoring of user behavior to see just which new features are used, and how they are used, thus becomes another required core competency."[6]

Since the rise of Web2.0 new opportunities for testing, distributing and updating their products are available for technology and software developers. Though stemming from a linear engineering-perspective on software development processes, the emerging trend of democratizing communication structures on the Internet has influenced the work of professional system designers as well. The developers arm now can reach into the living (and working) environment of users via the Internet. Thus, systems have become open for bug fixes, changes and updates at almost any time after the installation of a product. This allows agile software development processes to let users test prototypes in earlier phases of development, in turn enabling developers to better adjust the product to its later use context. Open Source solutions like Eclipse rely heavily on a principle that Erich Gamma described as "always beta" [7] in an ICSE keynote. O'Reilly uses the term 'perpetual beta' to describe a similar issue [6]. The beta-ness, hence, does not refer to stagnancy. Instead the status just recognizes that innovations are based on the co-evolution of technical artifacts as well as the social practices that have emerged from its appropriation [8]. From this point of view, perpetual betas are a way to foster appropriation processes to be able to gain insights on and address upcoming user needs in better ways [9, 10].

This allows for a specific set of methods for development processes to be used, where households as well as designers and other stakeholders each have their place in the design process in which hard- and software can be tested. In the following we will show, how we dealt with these challenges using a Living Lab approach.

Designing an always-beta HEMS

The work described in this paper was conducted as part of a 3-year long project focusing on the research and development of concepts and strategies for in-house information systems, including HEMS. Our work started with a pre-study that included empirical studies on current energy consumption in participating households. This study helped us understand basic energy practices and we gained a first impression of the HEMS design space and the users' needs [11]. In order to deepen our understanding we started to set up a Living Lab infrastructure in seven households in early 2011.

In general, the Living Lab approach [2, 12] is aimed at bringing users and technology into an open design process in a real life environment [12]. The concept supports long-term cooperation, co-creative research and development by involving, at an early stage, the user in the design process for 'sensing, prototyping, validating and refining complex solutions in multiple and evolving real life contexts' [2]. The long-term cooperation between researchers, users and other relevant stakeholders distinguishes this concept from other approaches.

Coping with the challenge of innovating design, we regarded the living labs as a greenhouse, in the way that novel design concepts can grow within a fertile environment to more robust and comprehensive solutions in an always beta process. In particular, by creating opportunities for users to have first-hand experiences is useful to address the complexity and situatedness of HEMS use in real-life environments.

Setting up the always-beta infrastructure

In the case of Smart Homes, the beta-ness not only covers the software artifacts, but we also need to consider the home and the household itself as a beta infrastructure. In particular, making a home "smart" usually implies the rollout of new hardware including new media devices, new sensor technology, and new home network components. In this section, we will outline these special conditions and its implications for our research.

Given the future prototype would consist of both hardware and software components, it was important to define the further general hardware specifications in a very early phase, to gain clear requirements for household acquisition. First, in order to create a basic infrastructure that was the same in all households, we decided to provide participants with a Smart TV connected to a Media Center PC.

The sensing infrastructure consisted of a Smart Meter to include the total household consumption data into our HEMS. Furthermore, we used Smart Plugs, which can be placed between the socket and the plug in order to sense energy consumption on the device level. Next to general requirements such as having an internet connection, choices on hardware had implications on our household acquisition. For example, since Smart Meters may only be ordered by the owner of a home, we had to spare out tenants.

Since the Smart Plugs are able to create ad-hoc multi-hop networks, they allow for an easy and flexible installation and can be placed wherever the user wants to measure the energy consumption of a device. Both installation processes needed the participants' active involvement and constituted an intervention into the privacy of their homes in a very early phase of the project. First, for the installation of a Smart Meter an appointment with an electrician had to be made. Second, together with the researchers, the Smart Plugs had to be rolled out.

We found it extremely useful, to in part rely on proven working technology (such as ZigBee and Smart Meters) in order to test the combination of these working components with newly developed hardware and software development. Therefore, where knowingly working technology exists, it is reasonable to make use of these components when building a prototype for real life environments in order to reduce the complexity of the system.

Monitoring emerging practices and ideas
We will now present our impressions and experiences when co-designing in real world environments such as private households and how we coped with challenges in the process of our project.

Often, when talking with the participants they expressed very blurry ideas of how to tweak the HEMS in their favor. For example, we were approached by the households with individual wishes concerning special functionalities of the HEMS without participants being able to define a corresponding visual representation of their idea. In each field trail, we gathered a manifold of these impressions, which had to be conserved. Therefore, it was important to contemporarily analyze and operationalize them into our goals. By doing so, during the re-implementation of the HEMS prototype, the context of ideas appearance could be recalled to better understand the motivation and the aims of the data representation desired. We found, that confronting participants with a new prototype helped them sharpen and refine their hitherto blurry ideas by interacting with the HEMS, thus finding new bonds between system and their strategies. It obviously helped participants to be able to experience what it really meant to use such a system in their own household, to fully explore its possibilities.

Additionally, we tried to get a description of what HEMS features were used in what context, what their ascribed meanings were and what features were further needed. For this we profited from the fact that our settings allowed the triangulation of interviews, log data and site visit observation. Using log data analysis, we found that live consumption data was the preferably used resource as opposed to historical data. This largely corresponded with our impressions from the on-site visits, where participants often wanted information on their current consumption, in order to understand which devices contributed to the amount of energy used momentarily. On the other hand, our talks showed how participants also wanted support in reflecting on historical energy consumption independent from time and space, therefore needing contextual information in order to recreate usage situations. When we designed such a support, and site-usage of the historical feedback page did not improve, this was a hint for us that the system had not sufficiently provided support for households´ energy practices. Overall, the broad data acquisition was very helpful for monitoring emerging practices and reflecting on system design.

Another issue that the user research has to consider is that through the long-term cooperation participants somehow become always-beta user experts. In particular, they in part developed a certain frugality which seemed to have influenced their open innovative thinking. Furthermore, after two years of co-designing, the households might have also become too used to the prototype to spot problems in usability, for example, because they became used to them.

Updating the infrastructure

When designing in the real world, the surrounding infrastructure holds a manifold of possible obstacles, especially for hardware. For example, our ZigBee based sensor network showed very varying performance. This was due to the building material used. In one case a lot of steel and iron was used which reduced the transmission power heavily. Since our Smart Plugs allowed for plug-and-play extension, we were able to strengthen the network by deploying more sensors with a smaller distance in between them. In other cases, people wanted to host their HEMS on a different device than the provided media center PC. This was possible if the operating system was at least Windows Vista or a newer Microsoft OS. The provision of feedback based on host data was possible from any device in the home area network. Currently we are aiming at migrating the host to a Linux-based mini-computer, in order to provide host data permanently. This is easily possible, since the infrastructure is built modularly and uses established standard interfaces. Designing the hardware-prototype with this in mind, updating the infrastructure becomes less stressful due to the initial efforts to build the infrastructure modularly.

Additionally, we also profited from the trustful relationships we established (see bellows). Based on these relationship we were able to make appointments for remote connections to the HEMS more flexibly and via more informal ways of communication such as private, workplace or mobile phone numbers. Households also allowed us more and more to connect to their computers and routers, even when they were not at home. This helped the development process in terms of test and bug fixing efficiency.

Being a trustful help desk in long-term cooperation

In the course of the project, we found that participants increasingly perceived us as a helpdesk for all technological concerns. For example, one participant asked us how to update his router firmware, while another asked us about the features of his new laptop. Though this sometimes ostensibly appeared to be a waste of time, the questions may be seen as an expression of the research team being seen as trustful technological experts. Refusing to adopt these tasks might have had wider consequences for future co-working experiences and might have seriously hindered the establishment of the trust-relationship between designers and participants.

In particular, when dealing with prototypes in complex environments, system failures are unavoidable. The households were well aware that our designing aims were very challenging and therefore were very understanding when the system did not work properly. A crucial aspect though, was the credibility of the provided feedback data. In case feedback did not make sense to the participants or conflicted with their beliefs, the negative user experience revealed long-lasting negative effects on system use. The system had to be trustworthy for the users. We concluded that showing no information was the lesser of two evils, compared to providing false feedback. Otherwise this had negative effects on the motivation and use of the prototype from which it would be very difficult to recover from, even in the long term.

A key factor for making amends for possible system failures or wrong information was the establishment of a trustful relationship between households and designers. Even more, without the openness of the households, telling us their thoughts and problems with the material "energy consumption", we would not have been able to get insights into their daily practices. Therefore it was very useful to assign each household a contact person from the team who was responsible only for that household. When visiting people in their homes, listening to them and asking about their energy relevant knowledge and spending time together while conducting workshops at the same time, the researchers were able to get even closer to the participants. But the positive relationship did not just develop by itself. We found it very important to always keep appointments and make the participants feel that their concerns were heard and were regarded as important, in order to allow trust to be built. Asides from being crucial for understanding their individual use of the system, especially including the difficulties in managing energy consumption as a phenomenon for design research, the trustability had various other effects on the process. On the one hand, the long and trustful relationships made it easier for us to update the systems (see above), on the other hand, when people forgave us delays or system failures, it also meant that they had created their own workaround and learned to live with a provisional arrangement. Both issues have to be taken into consideration when generalizing user practices (see above).

Discussion and Lessons Learned

In this paper, we have presented our experiences with adopting an always beta philosophy within a Living Lab approach in the domain of designing hardware and software for the Smart Home. We reported on the challenges with the technology and with the participants and how we coped with them.

Designing for and with private households in an always beta approach, we were faced with two major challenges. Both are grounded in the very nature of the methodology: the early testing of prototypes "in the wild". In the following, we will summarize our impressions and explain how we value these challenges for developing in an always beta philosophy.

First, trust is a major requirement for a successful co-design process. When working with prototypes, a system might show its limitations in certain home environments, or even fail to work or at least not work the way it was expected by the user. Additionally, only when being trusted, participants might open up and express their feelings, fears, problems or limitations regarding their skills and understanding of the system interaction. For these reasons, building a relationship of trust with the user and making them understand the prototype-character of the provided systems, is key to being able to repeatedly change design and fix bugs without households becoming impatient or losing trust in the system. Furthermore, especially since the prototype intervenes in their private sphere, acceptance of the researcher and the system go hand in hand. It is important to make the participants feel that they are being heard and respected, in order to make them tolerate problems . Going the extra mile is worth it. As a reaction to participants being less critical and starting to get used to HEMS, we are currently searching for new participants to re-evaluated the developed system with unprejudiced eyes.

Second, the heterogeneity of the field of application calls for special attention. Households rely on such a wide variety of different technologies, living situations in terms of people and space, surrounding infrastructures and building material, that anticipating them or even their meaning for design to their full extent is impossible. While software does not necessarily rely on these factors, when designing hardware in the household, all of these variables may possibly influence the functioning of a prototype. Used hardware should therefore be flexible, modular and expandable to be able to use more powerful network-technologies, for example, when needed. Moreover, if possible, it should be made use of established technological standards in order to reduce the complexity and error-proneness of the system itself thus being able to focus on the software-development.

In conclusion, though demanding in terms of setup and maintenance, we believe an always beta philosophy can provide a unique added value when developing combined hardware-software-solutions. The outlined challenges, when handled well, exactly constitute the added values that an early phase rollout of hard- and software in real life environments can provide: being able to make use of the impressions from use in the complexity of the users' domain in a design phase where many decisions can still be adjusted to the experiences of technology appropriation in the households. By doing so, the gap between existing user expectations and the system can be reduced, by continuously updating and improving the system in the design, test and evaluation loop. Following this principle, we were able to set-up our prototype to match practical infrastructural needs and reflect the users evolving appropriation strategies.

In the future, we plan on continuing the always beta process and to adapt it for other usage scenarios of our HEMS such as mobile usage. We are currently building a mobile extension prototype for HEMS, which will be further developed within our Living Labs.

Acknowledgements

We would like to thank all the participants for enabling us to gather insights on their daily routines and all their innovative and challenging ideas.

References

[1] Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131-133.

[2] Eriksson, M., Niitamo, V. P., & Kulkki, S. (2005). State-of-the-art in utilizing Living Labs approach to user-centric ICT innovation-a European approach. *Lulea: Center for Distance-spanning Technology. Lulea University of Technology Sweden: Lulea. Online under: http://www. cdt. ltu. se/main. php/SOA_LivingLabs. pdf*.

[3] Bernhaupt, R., Obrist, M., Weiss, A., Beck, E., & Tscheligi, M. (2008). Trends in the living room and beyond: results from ethnographic studies using creative and playful probing. *Computers in Entertainment (CIE)*, *6*(1), 5.

[4] Schwartz, T., Denef, S., Stevens, G., Ramirez, L., & Wulf, V. (2013, April). Cultivating energy literacy: results from a longitudinal living lab study of a home energy management system. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1193-1202). ACM.

[5] Jakobi, T. (2012). Integrierte Heim Energie Monitoringsysteme (HEMS) für iTV: Eine LivingLab basierte Design-Fallstudie. Master-Thesis.

[6] O'Reilly, T. (2009). *What is web 2.0*. O'Reilly Media.

[7] Gamma, E. (2005, May). Agile, open source, distributed, and on-time: inside the eclipse development process. In International Conference on Software Engineering: Proceedings of the 27 th international conference on Software engineering (Vol. 15, No. 21, pp. 4-4).

[8] Stevens, G., Pipek, V., & Wulf, V. (2009). Appropriation infrastructure: Supporting the design of usages. In End-User Development (pp. 50-69). Springer Berlin Heidelberg.

[9] Ullrich, C., Borau, K., Luo, H., Tan, X., Shen, L., & Shen, R. (2008, April). Why web 2.0 is good for learning and for research: principles and prototypes. In Proceedings of the 17th international conference on World Wide Web (pp. 705-714). ACM.

[10] Hoyer, V., Stanoesvka-Slabeva, K., Janner, T., & Schroth, C. (2008, July). Enterprise mashups: Design principles towards the long tail of user needs. In *Services Computing, 2008. SCC'08. IEEE International Conference on* (Vol. 2, pp. 601-602). IEEE.

[11] Betz, M., Schwartz, T., & Ramirez, L. (2010). Know Thyself: Monitoring and Reflecting Energy Consumption.

[12] Følstad, A. (2008). Towards a living lab for the development of online community services. The Electronic Journal for Virtual Organizations and Networks, 10, 47-58.Zellweger, P.T., Bouvin, N.O., Jehøj, H., and Mackinlay, J.D. Fluid Annotations in an Open World. *Proc. Hypertext 2001*, ACM Press (2001), 9-18.