

# Enabling User Centered Design Processes in Open Source Communities

Mads Bødker, Lene Nielsen, and Rikke N. Orngreen

Center for Applied ICT, Copenhagen Business School,  
Howitzvej 60, 2000 Frederiksberg, Denmark  
Mb.caict@cbs.dk, ln.caict@cbs.dk, rno.caict@cbs.dk

**Abstract.** Drawing on tenets from action research, this paper presents a yearlong intervention designed to facilitate knowledge of actual users and use in an Open Source Software (OSS) development community. Results from the interventions are presented and the influence of central characteristics of the OSS community and its communication is discussed. Initial findings show that the ideology and praxis based approach of the OSS community, as well as their primary media of communication, present a challenge to the introduction of end-user issues.

**Keywords:** Open Source, usability, developers, community, learning, action research.

## 1 Introduction

This study reports from a project that aimed at introducing usability and user awareness into an Open Source Software (OSS) developer community. In the field of human computer interaction (HCI) a variety of methods exist that focus on user centered design. A wide range of methods exists that investigates user needs and context as well as methods that involve users directly in the development process (participatory design) [7], but these have not yet been widely applied in open source development. This potentially leaves a gap between the developer-users, those who extend and innovate on the OSS, and end-users, those who will end up actually using systems for their intended purpose. The question for the following is, if user-centered design thinking is something that can enter into the Open Source development environment?

Existing usability studies in connection to OSS development focus on how users report bugs and wishes for new systems features as well as how the development community reacts towards these reports and wishes [2, 10]. Other studies on user involvement in design processes often focus on user driven innovation, where involving users in the development process using a variety of methods are claimed to bring about new and innovative designs [6]. In particular of interest to the open source community are the so called Innovative communities [5]. As such, the OSS development community qualifies internally as an innovative community – there is a strong element of competition and innovation in OSS development. However, the user

in the case of OSS innovative user communities are the before mentioned developer-users, not actual end-users of technology. In this way, innovation in the OSS community does not extend across the boundary into the context of those users who use the systems on a daily basis for mundane and leisurely task. Correspondingly, knowledge of end-user situations can not be seen to play any significant role in the development of the system and potentially not arriving at end-user friendly products can pose a serious threat to popularity and adoption of the system.

This paper introduces a yearlong intervention project carried out through most of 2006 with members of the Open Source community behind the Content Management System TYPO3.

## 2 TYPO3

TYPO3 is a widely used small to midsize enterprise class content management system (CMS) under an Open Source license. TYPO3 has been publicly available for 5 years, and it has currently approximately 320 active contributors.

The TYPO3 community has never signed up for a formal membership. Rather, it consists of people who join the TYPO3 mailing lists, newsgroups and more formal groups, for example the R&D group, the Core development group and so on. The members are a highly diverse bunch: some are highly skilled programmers who participate with an interest in developing system extensions; others are interested users who use the mailing lists to put questions about use. The community is organized in several subgroups and communication takes place in discussion lists as well as in occasional physical sub-group meetings. The discussions seen at the TYPO3 community (see the [typo3.org](http://typo3.org) website) are generally oriented towards the implementation of extensions to the system or which bugs should be fixed.

On the [typo3.org](http://typo3.org) website the R&D group, in its own words, state that their aim is developing a system which is complex and yet usable so as to support business CMS solutions. At one point, the team had chosen to address the issue of usability in their coming TYPO3 versions, but realized that the “code now, humans later” [15] focus of the developer community made it difficult for them to attract the knowledge needed. This problem made the R&D group approach the authors of this paper in order to initiate a process of introducing usability awareness to part of the community.

## 3 The Method

The research presented in this paper is based on principles and ideas from action research. The aim of action research is to create change by improving a specific case, in a specific period of time, at a specific location [17].

The action research label accounts for a number of different attitudes towards research process and methodology. While the change-oriented contention of action research is central, there is a great variety and no methodological canon to be followed. [1]. Our central aim in adopting an action research approach to the project was that we did not only want to study the state of the art in the developer community, but throughout the study maintained a therapeutic stance, wanting to change the orientation of the developer community.

Our project fell in two distinctively different phases that we have termed the Ambassador Project and the Learning Project. Both phases took place in the context of a HCI discussion list we had set up to officially indicate that an initiative directed at improving TYPO3 usability was in progress.

In the first intervention, the Ambassador Project, we as researchers would have to get an understanding of the users and use situations. Then we would ask the ambassador participants to investigate their users and share their knowledge with the other developers [12]. These were to be used as a basis for personas descriptions intended at the distributed development process.

In the second intervention, the Learning Project, we introduced a set of heuristics in order to provide the participants on the mailing list with a common vocabulary for usability, supposing that having some form of contextually relevant knowledge on usability, equally available to all developers, would set some form of reflection upon end-user issues in motion.

## 4 First Intervention: The Ambassador Project

As with most other OSS projects, the TYPO3 development structure consists of developers who carry out programming of the core TYPO3 system (the stand-alone system) and scores of developers in the various user-groups who use the TYPO3 source-code to program individual business solutions. In practice, developers may take on both roles – doing coding for the sake of the TYPO3 system itself and suggesting new features found usable during individual projects, suggesting them for implementation in TYPO3.

From following discussions on the TYPO3 HCI-list, it quickly became apparent that there was no explicit and common knowledge in the community of whom the actual end-users are. As part of the process of creating awareness of the users, we conducted a pre-study of the use of the TYPO3 CMS in two organizations differing in size, complexity of the TYPO3 system implemented, and the end-users' possibilities for IT support. Four interviews and three videotaped observations were made. Talking to end-users and seeing them use the system provided insights about work, work situations, and attitudes. Attitudes originated in computer skills with end-users being either comfortable with computers thus putting demands to the system or uncomfortable with computers, but pleased with the system as long as fixed procedures were followed.

Table 1 below shows an excerpt of a description of an actual end-user, based on a process of meaning condensation [2] from our interview with and observation of an end-user in a large public organization using TYPO3. Sarah, who is a highly skilled technological user, is focused on how the system fits her working processes and needs.

Descriptions similar to this, derived from our observations, were used to draft HCI ambassadors. Our assumption was that poignant examples of real, "lived" experience of TYPO 3 use could attract developers with particular interests in usability and end-users to act as ambassadors on the list. Coming from the same "programmer culture" as the other developers on the list, we assumed that these ambassadors would be better equipped to disseminate their interest to the wider community.

Sarah is a legal advisor, employed in a large Danish public organization and uses TYPO3 everyday. Her attitude towards the system is that it has not been designed to accommodate for her way of working and the tasks she has to perform. She has to scroll far too much among the documents and she has many ideas of how to improve the system.

Sarah gives an example of one of her tasks carrying out statistic logs of legal decisions, where she states that “this is a real nerd-calculator, top-nerd! ... so I can’t use this for anything” and continue to show how she performs the calculation manually. She has tried to speak to the IT department about improving their adapted version of TYPO3, but feels that they don’t understand her. As she explains: “They speak Chinese and we speak Danish and there seem to be no dictionary”

When she started using the system she spent some time on an in-house course, but she has mainly learnt the system by using it. This also makes her create her own shortcuts and “this is why I have developed my own ways of working with the system”, she says when pointing to some of the obstacles of the systems features. She is a person that others contact when they are stuck. Recently she spent two days training a newbie to TYPO3, as Sarah has been offered a new job. This was quite a frustrating experience and as she says in her new job: “I’m never going to play with TYPO3 again”

**Fig. 1.** Except from a description of an actual TYPO3 user

On the HCI list, we asked the developers to consider what they knew about their end-users, and to submit written descriptions of actual end-users they had met. They immediately perceived the request for user descriptions as a request for descriptions of abstract user-types, which they denoted “personas” in their discussions. Taking a solution-oriented approach the developers used these personas to describe solutions for the system. Later they were asked to interview a selection of users and four e-mail interviews were carried out. These interviews showed that most end-users were content with the system, but they also exposed a huge variation in the use of the CMS. Either it was used by novice users with a very limited set of functions on a less frequent basis, or by users with high computer skills, using a wide range of functionalities on a daily basis. This supports the observations made earlier, but the interviews were too few to be of any actual value. While the ambassadors were well versed in communicating on the HCI-list, they lacked knowledge on usability concepts and aim, and even if they found it to be important, it never became clear to them what the aim of the project was and no more data came out of it. This made us close down the project to continue along another line of intervention.

## 5 Second Intervention: The Learning Project

The correspondence on the HCI-list exposes a frequent inability to cope with engagement in end-user issues other than by implementing rapid solutions to clearly specified problems. To the developers, engaging in users is seen as a solution oriented problem, since end-users are perceived as solution finding actors. Taking over where the ambassador project ended, we decided to use the HCI list more systematically to

facilitate a learning process amongst developers enthusiastic about HCI. Primary in this process was the use of the mailing list and the TYPO3 wiki to assist the group in creating a shared vocabulary and provide clear examples of usability thinking that serve as guidelines for deliberating and subsequently solving problems.

The HCI-list has developed steadily since May 2006 and it still features a lot of discussions about solutions with a noticeable exception being a discussion taking place in late September 2006. A thread started by TYPO3 founder Kasper Skårhøj whose re-reading of an article [1] instigated the asking of more critical questions about end-users. The thread can be distinguished from others in the HCI-forum since it sought to determine which solutions are better considering end-users and the motivation of developers to solve end-user problems. While posts about specific solutions get more attention in terms of replies, this discussion occasioned a rather extended dialogue consisting of 24 posts from 11 different posters. However, the problem seems to be that discussions lapse towards either specific problems (e.g. labeling of functionality), towards paradigmatic observations of a very general nature (e.g. are we “dumbing the system down” or are we making it smarter?), or towards ethical paradoxes inherent in open source development (e.g. why care about users at all when you do things for free?).

Since our analysis of the discussions indicated a generally poor understanding of the concept of usability and as the community seems mainly to concentrate on technical solutions, we tentatively introduced 10 heuristics (see [wiki.typo3.org/index.php/Heuristics](http://wiki.typo3.org/index.php/Heuristics)) derived from [11], [14], [9], [16] and chosen amongst the many principles for design introduced by the authors to reflect typical problem areas in the specific TYPO3 CMS domain. With a detailed description of the heuristics posted on the mailing list and available to the discussants on the TYPO3 wiki, we were hoping to facilitate a shared vocabulary for the developers, a common place of reference enabling a process where problems with the TYPO3 interface were no longer seen as highly specific, but as indicative of more principal problem fields and hence applicable to heuristic analysis. A shared vocabulary and some basic knowledge of more abstracted concepts in understanding how users interact with systems, so we expected, would raise the bar for the discussions on the list and potentially make the problems discovered eligible for shared solutions rather than the unsystematic and narrow focus of solving “one-off” particular problems.

## 6 Analysis

Introducing concepts of usability and, more broadly, an understanding of- and empathy with users into the OSS development community, proved to be a challenging undertaking. In the following we will analyze and evaluate how the community of TYPO3 developers interacted and how our intervention was used in the community.

Since the development of TYPO3 is Open Source, we find it necessary to look at a network of OSS discourse, ideology, and praxis to see how these can be said to conflict with our style of intervention and the possibilities for change and learning. Further, we will assess the medium wherein learning and communication was facilitated.

## 6.1 OSS Discourse, Ideology and Practice

Even if the project did succeed in setting up an active list for HCI interested developers (the HCI list), a common objection on the mailing list was “why should we develop for “users” (meaning here end-users) since what we do is essentially for free and since we do it simply because we like programming, why should we care how or how well “regular” users use the things we build? For instance, HCI-list postings along the line of “*a core developer has no responsibility above whatever his personal motivation may be*” (Oct. 13. 2006) or “*why is it not naturally for everyone to scratch ones itch?*” (Oct. 12. 2006). This line of reasoning is reminiscence of what we could call classic OSS discourse or ideology. Using a concept of ideology as a normative structure that tacitly and seemingly a-historically allow us to think and believe in specific ways, OSS ideology seems to rely strongly on classical democratic tropes of sharing and equal relationships between peers. Sharing and transparency is key terminology used to describe the nature of a working OSS development community [8].

Yet since OSS is by definition developed “con amore” and with no direct economic incentives, there is no perceived obligation to actually “care for the itches of others” - to have any kind of empathy for those outside the loosely coupled group of developers who share knowledge, skills, values and vocabulary. As Eric Raymond, who co-coined the term Open Source states in his seminal book “The Cathedral and the Bazaar”: “Every good work of software starts by scratching a developer’s personal itch” [13]: p. 23.

Therefore sharing and transparency are attributes that are at work within the community of developers themselves, not something that has any relation to end-users. In short, we can say that the Open Source encouragement structure and the non-hierarchical community arrangement and the strong sense of emotional belonging that the community commands tends to preclude the possibility of seeing beyond their own motivations. Thus while the originating ideology of Free and Open Source software development seems to hinge on an altruistic, purified democratizing effort and a cleansing of capitalist incentives in the development of technology for the coming Information Society, there is a marked element of neo-liberalist thought that disqualifies the perception of ones work as a service to end-users, those who are not themselves part of the development community. Hence, the community ideology itself presents a challenge to the introduction of user-centered thinking – there is simply no obvious incentive.

## 6.2 Praxis

Another aspect that posed a severe challenge to our intervention was that while our project sought to activate the developers reflexivity of other peoples use of their product in order to facilitate more user-centered design, OSS culture is, as De Joode has pointed out, a culture of *doing* not of *deliberating*, a specific communal trait that we will also discuss in the next section. OSS development is a culture of proving ones

worth in practice, not of using abstract ideas to guide ones practice. OSS development is a zero-sum game where the provably best piece of code is adopted into the system while the less functional ones are abandoned [18]. Compared to that, user-centered design, even when taking up empirical modes of inquiry is quite different when it comes to proof or proving. While user centered design experts can make educated assertions about how interaction will take place, and hence derive a set of heuristics from these assertions, a user-centered product is not static in the same sense that a provably more efficient algorithm is. This difference in the culture of proving and the praxis derived from it could be seen as one of the reasons for the failure of our initiative to introduce principles of usability to the community.

### 6.3 Communication – The Medium

The primary coordination and communication tools used in OS development communities are e-mails, mailing lists, forums, and other forms of digital networked communication tools [8]. So too, in the TYPO3 HCI community where the primary communication took place on the HCI mailing list. While the TYPO3 development structure does have a certain hierarchical organization, using an appointed association to take decisions on official TYPO3 releases and certification of commercial TYPO3 agencies, no hierarchical organization existed to enforce decisions or to evaluate the outcome of the ongoing discussions on the mailing list. As such, the list provides a space for a kind of *ideal speech situations* [4] where no external criteria are used to evaluate the rationality of communication. However being principally unperturbed by external power also disabled decision making, as the discussion on the mailing list itself was not able to make decisions that cut across group boundaries, for example across those who are in favor of making the interface less complex and those who favor extensive end-user programming. Acting mainly as a pure discussion forum, no efforts were put into enabling e.g. consensus decision making such as it is carried out in other open-standards and open source organizations (e.g. W3C, see [19]). As we have seen, this resulted in the list being used to primarily share specific solutions to specific design problems, and to share immediate problems, which could potentially be solved by other participants on the list.

While the power-free communication of a mailing list could be said to be a decent and indeed moral procedure highly consistent with the OSS community ideology, it did not, in our case, meet the criteria of actually pushing the innovation of TYPO3 usability forward. Since it was mainly used to assist in the solving of individual users problems, and hence enabling them to better “scratch their personal itches”, our analysis shows us that we did not adequately assess the problems inherent in using the mailing list to facilitate usability learning and subsequent innovation. Innovation here should be understood as the introduction of a considerably more user-friendly interface in coming official TYPO3 releases. One way to solve this problem of on-line, distributed decision making could have been to facilitate consensus based decision making, entailing for instance that propositions should be clearly marked as such, thus being eligible for assessment and consensual verdict, or by appointing a managing committee to enforce some form of conceptual integrity [3] – making sure that the hundreds of ideas that surface on the list correspond to one overarching goal.

## 7 Conclusion

OSS development, while proceeding directly from the practical and technical tenets of software engineering, has since the early days of hacking departed from traditional management principles such as those presented by Brooks in his book on the Mythical Man Month [3]. Where Brook's assertion was that adding more programmers to a delayed project slowed down development, Eric Raymond argues that the OSS strategy of distributed development has other virtues. Certainly, as he argues, "given enough eyeballs, all bugs are shallow", indicating that distributed bug elimination, loosely coordinated via the Internet, is indeed an efficient strategy. The question is if the same anarcho-libertarian OSS tactics are as efficient when it comes to designing for "real" end-users?

A number of things came of our intervention. First of all, the fact that issues pertaining to usability are explored and discussed at all. Since the personal motives in OSS development are primary to all other incentives, this can be seen as an innovation in the TYPO3 community. Secondly, the participants on the list began as of late November 2006 to work on a comprehensive survey to assess how users experienced the usability of TYPO3. The real innovative aspect of this survey was that it was highly attentive to the fact that there were indeed many kinds of users, and that, in order to communicate and gain insight into a multitude of users, different kinds of language should be used and different kinds of questions were needed. Before our intervention we found the development team framing their discussion of users as a discussion of the conflict between designing for "dumb end-users" or for highly skilled "administrator-users". The survey and the associated discussion on the list suggest that our intervention pushed some developers' attitude from antipathy towards empathy with end-users.

A pertinent lesson to be learned from our inquiry is that the developer segment in OSS is not particularly disposed to concern themselves with phenomena outside of the community. Rather than addressing the developer segment, it might be conceivably more sensible to address the actual user segment. Enabling the organizations and institutions that make use of OSS software to understand their employees and enabling them to specify the right requirements for their system, might be a more efficient way to avoid developer-centric systems that perform poorly in terms of real-world usability.

## References

1. Baskerville, R., Wood-Harper, A.T.: A Taxonomy of Action Research Methods. PAP012-0.05 (1996)
2. Benson, C., Müller-Prove, M.: Mzourek, Jiri.: Professional Usability in Open Source Projects: GNOME, OpenOffice.org, NetBeans. CHI2004. ACM, Vienna, Austria (2004)
3. Brooks, F.P.: The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley, Reading (1995)
4. Habermas, J.: The Theory of Communicative Action. Beacon Press, London (1981)
5. Hippel, E.V.: Democratizing Innovation. MIT Press under the Creative Commons Rights (cc) (2005)



6. Jeppesen, L.B.: User Toolkits for Innovation: Consumers Support Each Other. *Journal of Production Innovation Management* 22, 347–362 (2005)
7. Kensing, F.: *Methods and Practices in Participatory Design*. ITU Press, Copenhagen (2003)
8. Ljungberg, J.: Open Source Movements as a Model for Organising. *European Journal of Information Systems* 9, 208–216 (2000)
9. Moore, P., Fitz, C.: Gestalt theory and instructional design. *Journal of Technical Writing and Communication* 23(2), 137–157 (1993)
10. Nichols, D.M.: Twidale, M.B.: The Usability of Open Source Software. *First Monday*, vol. 8, 1 (2003)
11. Nielsen, J., Molich, R.: *Heuristic Evaluation of User Interfaces* (1990)
12. Nielsen, L., Orngreen, R., Nielsen, J.: *Engagement in Users - a new approach to open source development*. *International Design for Engagement*, Oslo (2006)
13. Raymond, E.S.: *The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*. Revised edn. O'Reilly & Associates, Inc., Sebastopol, CA (2001)
14. Shneiderman, B., Plaisant, C.: *Designing the User Interface*. Pearson, Harlow (2005)
15. Skårhøj, K.: TYPO3 - presentation to HCI students. E-business, IT University, Copenhagen (2005)
16. Tognazzini, B.: *First Principles of Interaction Design*. AskTog <http://www.asktog.com/basics/firstPrinciples.html> (2003)
17. Toulmin, S., Gustavsen, B.: *Beyond Theory. Dialogues on work and innovation*, vol. 2., John Benjamins, Amsterdam (1996)
18. Van Wendel De Joode, R.: *Understanding Open Source Communities - an organizational perspective*. Technische Universiteit Delft, Delft (2005)
19. W3: <http://www.w3.org/2005/10/Process-20051014/policies.html#Consensus>