Best Practice for Efficient Development of Inclusive ICT

Till Halbach Røssvoll and Kristin Skeide Fuglerud

Norwegian Computing Center (Norsk Regnesentral)
Gaustadalléen 23, 0314 Oslo, Norway
{Halbach,KristinS}@nr.no

Abstract. This work aids ICT projects in delivering solutions which are highly accessible and usable in an efficient manner, i.e., with a minimum of additional costs. The aid comes in the form of guidelines / best-practice recommendations. The guidelines are based on a literature survey considering related research, and an analysis of development work conducted at our research institute which discusses possible pit-falls. This approach led to both high-level recommendations, such as which overall research methodology to apply, as well as detailed low-level guidelines, such as which activities to include in the project workflow and when. The advice is supported by a template for an example workflow with relevant activities. The recommendation from the literature is to extend general user centered methodology with particular activities to ensure that also individuals with impairments are accounted for, while our own development experience suggests an iterative approach with user involvement from early on in the project throughout the end.

Keywords: Software development, agile user centered design, IT, ICT, universal design, e-inclusion, accessibility, usability, impairments, iterative design, best practices, workflow.

1 Introduction

Since the introduction of the Agile Manifesto in 2001 [1], many areas in software development have undergone a development turning from traditional methods like waterfall towards methods that promise a better cost/effort ratio, such as Lean, eXtreme Programming, Dynamics Systems, Rapid, Iterative, Test-Driven, and others. These methods typically promise a reduction of the project risk and consequently savings of project expenses. Starting more than 10 years earlier [2], many software projects have been witness to another evolution step, with a turn towards user centered design (UCD) processes. Several factors have contributed to this recently, including the success of well designed Apple products, the open-source community with high degree of user engagement, the erection of Living Labs, and the participatory-design movement, to mention some of the most important influences. This development has eventually been acknowledged by documenting human centered design by multiple ISO standards, including [3].

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2013, Part I, LNCS 8009, pp. 97–106, 2013.

It has, however, been pointed out in the literature that the influence of the UCD movement on agile methods is rather limited [4–6], as agility often is programmer and not user centered [7–9], and thus not helpful in increasing a solution's accessibility and usability. Also, a great deal of projects are aware of supposed additional costs for universal design, at least in a suspected short-term view [10]. This picture is diversified by projects being aware of a wide range of costs of various UCD methods, where for instance an informal expert review is considerably cheaper than a field study.

Cost and efficiency are both important factors. Naturally, UCD is a bigger effort than, say, self- or ad hoc design. Moreover, the design community has warned against testing of every single design decision and hence overtest solutions [11]. More knowledge is therefore needed on how to efficiently apply UCD methods in relation to inclusive design in order to avoid unnessecary expenses.

This work reviews related research regarding recommendations for the efficient development of inclusive ICT solutions and also rates a number of own development projects. Applying the resulting best practices is supposed to lead to a high degree of accessibility and usability of ICT solutions given a particular budget and time frame.

This work is funded by the European AAL Program as part of the MobileSage project [12]. It is structured in a straight forward manner: First, we review related research and summarize recommendations from the literature. We then assess development projects conducted at our research institution, before we present a list of best practices based on a discussion of our findings. At the end of this work, the conclusion is drawn.

2 Recommendations from Related Research

There are several inclusive design approaches which share a common goal of making technology accessible to as many people as possible, including people with disabilities. While there is broad consensus that it is good practice to follow accessibility standards and guidelines in inclusive design, it is an increasing awareness of the fact that this is not enough to achieve inclusive ICT-solutions. The general recommendation is that inclusive design must be based on a user-centered design (UCD) process which includes user testing with disabled people. In the following, some central features of various inclusive design appraches are briefly described.

The concept of Universal Design was introduced in the mid-eighties by the architect Ronald Mace, and has since then been adopted in many fields, including more recently in the design of ICT [13]. The Universal Design process should be holistic, and this requires the participation of disabled people [14]. The emphasis is on counteracting unnecessary special solutions and to encourage equality. The seven principles of Universal Design represent the main qualities that a universally designed solution should fulfill [14].

The focus of Universal Usability is to design products so that they are usable by the widest range of people operating in the widest range of situations as is commercially practical [15].

A central part in Inclusive Design is to define and quantify the level of design exclusion in a product [16]. This is done by considering the spread of user functional capabilities (physical, sensory, cognitive) across a population. The basic assumption in this approach is that products exclude users because their features do not match user capabilities. By identifying the most excluding features in a design, they can be re-designed to be more inclusive [16].

In Design for All [17], the following areas are pointed out as important challenges: the need to include social context in analysis, the need to facilitate involvement of users with disabilities, the support of requirements engineering methods to facilitate the elicitation of requirements from different user groups, and the investigation of multimodality [18].

In User-Sensitive Inclusive Design [19], it is argued that greater empathy with users is needed, and that this must be reflected in design methods, research, and development. By using the term user-sensitive as opposed to user-centered, they want to communicate the lack of a truly representative user group, the difficulties of communicating with users, ethical issues, and the importance of the attitude of designers [19]. In this approach, it is suggested to consider a number of specific "extraordinary" users in-depth and design for them in particular. These "extraordinary" users are elderly and disabled users which also have characteristics that are particularly relevant for the ICT solution in question.

The emphasis of Ability-Based Design [20] is to base the design on what abilities people have rather than what disabilities they have. The focus is on making mainstream technology flexible enough to meet the needs of people with diverse abilities through adaptability or adaptivity.

More and more researchers agree that compliance with accessibility standards and guidelines are not enough to achieve inclusive design. Recent empirical studies have found that conformance with WCAG can solve up to about half of the problems encountered by users with disabilities [21, 22]. As a consequence, some researchers make a distinction between technical accessibility and usable accessibility [23, 24]. In order for an ICT solution to be inclusive, it needs to be both technically accessible and usable accessible for people with disabilities. A solution is technically accessible if it provides access to content and functionality for people with disabilities. This can largely be achieved by meeting technical criteria in the underlying code [24], and many such criteria can be found in guidelines and standards. An ICT solution has usable accessibility if it is usable by people with disabilities, and this can be achieved by applying UCD and usability techniques, and involving people with disabilities. It is well known that early focus on users and their tasks, and early user involvement is important in UCD. According to [25], it is much cheaper to apply accessibility work from the start of a project than applying them near the end.

We should mention, though, that not all methods and techniques in UCD are always suitable for people with disabilities. For example, working with paper prototypes has been found to be an inexpensive and useful method to involve various users early in design. While this method is found to work well with elderly [26], it is not particularly suitable for blind participants. On the other side, while evaluation of working prototypes with the thinking aloud method has been found to be useful for many different user groups, including visually impaired and dyslectic users [27, 28], this method is difficult for elderly, possibly because it puts too high demands on their short-term memory [26]. It is thus necessary to consider what design techniques are appropriate for which users in what contexts and parts of the development process [16].

Empirical evaluation studies involving disabled participants usually find many problems that can be labeled as usable accessibility problems. Examples include issues related to navigation, ambiguous language and labeling, and inadequate help features [28, 29]. The same problems can often be observed across various types of disabilities [27–29], and will often be encountered by people without a disability as well, although non-disabled might not always be so seriously affected [24]. This means that evaluations with disabled people can be an effective way of uncovering general usability problems as well as for uncovering accessibility problems.

In summary, the overall recommendation from related research is to base the inclusive design process upon UCD, to complement with activities for ensuring conformance with accessibility standards and guidelines, and to extend and accommodate UCD methods and techniques to tailor people with disabilities. However, there is a need for more specific knowledge on how to handle the issue of diversity of users when moving from UCD to inclusive design [16, 30].

3 Experiences from Own Development Projects

Here, we summarize the experiences from development projects conducted at our research institute. These are as follows:

- DIADEM – a prototype for delivering inclusive electronic forms for impaired individuals and elderly [31]
- UNIMOD – a prototype for inclusive access to public services [32]
- KogNett – web pages for people with cognitive deficiencies [33]
- e-Me – a universally designed bill sharing application in social media [34]
- MobileSage – a smartphone application for context-aware adaptive guidance for elderly [12]

The DIADEM project had an effort of roughly 3.2 MEUR distributed over 36 months and involved 8 consortium members from three European countries. Its deliverable was developed on the basis of a spiral process. One development cycle consisted of the steps documentation (specification writing), implementation and integration, and evaluation. Unfortunately, multiple specifications (user requirements, system functional requirements, integration requirements, etc.) provided synchronization challenges to the programmer. It is deemed as beneficial to collect all implementation and integration requirements in a single document for the sake of simplicity. Next, the requirement specifications were mainly textual and not very specific when it comes to the user interface. It is thus recommended to include mockups and wireframes in the technical specifications to aid the programmer in the decision finding process. Last but not least, early design drafts were evaluated by experts but not end-users, resulting in a situation were the first-iteration user trials uncovered a great number of necessary major changes in the user interface, which were expensive to implement. Consequently, we recommended to let end users find poor design decisions in the early drafts.

UNIMOD was in order of 1.7 MEUR over 28 months, with 9 national partners. The project deployed an iterative design method, with a number of small-step integrations. This approach was chosen such that the participating public authorities could modify their requirements at any time. Small development iterations alternated with expert evaluations, while user awareness was honored by applying scenario and the persona method. As with DIADEM, early multiple designs were not evaluated by the end-user but rather expert users, which led to insecurity concerning if the chosen design would fit the user's needs in the end. We also learned that a great number of iterations may require a great effort by the programmer, as many aspects of the solution can be altered in each cycle. The recommendation here is to restrict the evaluation in particular iterations to only parts of the solution to reduce the programmer's effort required to implement all changes.

KogNett was a small-scale (50 KEUR) project over 2 months with only 2 consortium partners. Due to its size and limited duration, it deployed a waterfall method. User centered development was mainly accounted for in terms of several development iterations with expert evaluations, while testing by the end user came not until the deliverable's final public release. In developer terminology, the developing stages Beta and Golden were merged to a single deliverable, and flaws were fixed also after the final release. This approach worked surprisingly well, mainly due to the project's simplicity. However, until the final delivery it was unclear if the project really would meet the needs of the target group as the initial user needs analysis had been conducted by experts. It is therefore recommended to quality assure any expert needs analysis by an evaluation with users from the target group.

e-Me had an effort of roughly 1.7 MEUR over 33 months and involved 9 consortium members from a single country. It was based on rapid prototyping. While the development was to a great deal ad hoc based, it was evaluated in infrequent expert tests, followed by a final user trial to verify results. This pattern let the prototype run into the danger of not converging into a satisfactory solution. It is thus recommended to limit the number of evaluations to a well defined value. The lack of an up-front design resulted in a higher degree of redesign effort in the long run. On the other hand, decentralized decision making led to quick code (but not necessarily optimal) changes.

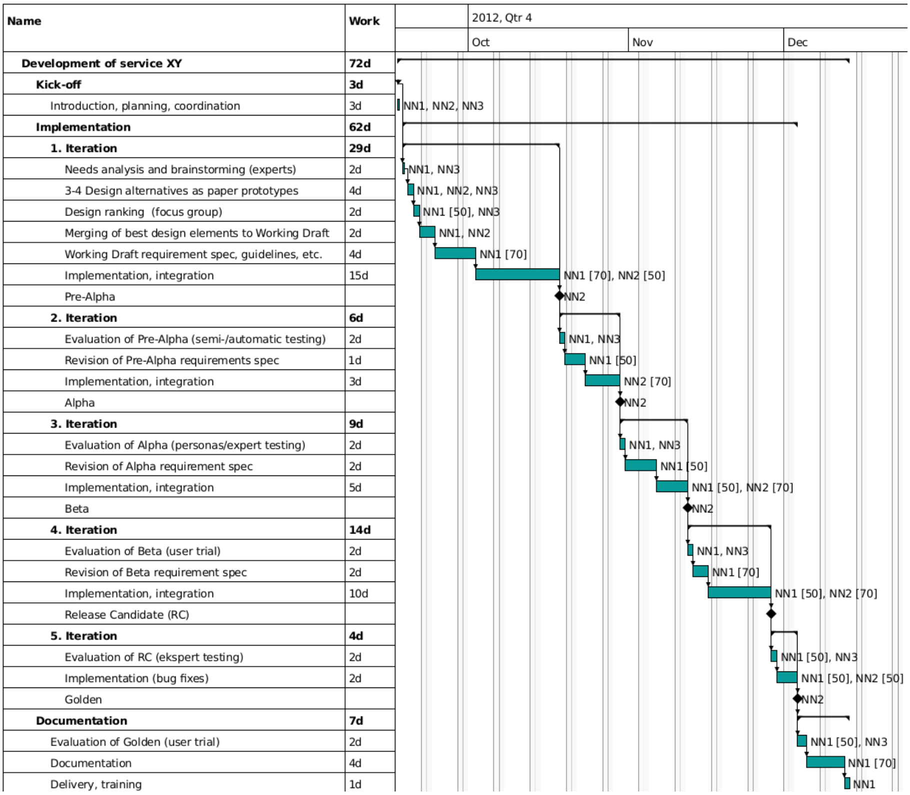
MobileSage is a project in the order of 2.8 MEUR, has a time span of 30 months, and involves nine members from three European countries. It utilizes a spiral method, with in total three development cycles. Each iteration consists of the development steps planning, implementation and integration, and evaluation. The initial planning step involved focus groups, and the last evaluation will be the verification of the final design. Other than that, each evaluation is based on the participation of users of the target groups. The spiral approach worked well, except that the initial focus group work was unable to discover some requirements that were viewed as important later on. It would thus make more sense to let an expert group conduct this step and have a focus group quality assure their work later on. Also, having user trials in each cycle is a cumbersome process which introduces some unplanned delay. Here, a number of issues could have been caught by an informal expert assessment in a more efficient manner.

4 Hands-On Best Practices

Few development methods foresee the involvement of users, be it traditional or agile. This must hence be handled by surrounding processes and/or the underlying framework. However, the majority of such frameworks, as for instance defined by the Living Labs movement, does not specify the exact method but rather suggests several alternative methods and techniques for user involvement, which must be selected on a best-fit basis for a given project [35].

Based on the aforementioned recommendations from the literature and our own development projects, we present the following best practices for an effective development of inclusive ICT. They are given on a general basis, to applied wherever possible. The user centered design activities are also shown in Figure 1 on the facing page, which can be used as a template for planning UCD.

- Have a user needs analysis brainstorming by experts to collect user requirements. This is cheap and accounts for system requirement as well.
- Prepare multiple designs in a co-design phase [36].
- Prefer paper prototypes over web mockups and grid building tools / wireframes, if possible, due to efficiency and cost issues.
- Sort out poor design and thereby merge good design decisions into a single Working Draft by means of a focus group with users from the target groups. With other words, combine parallel development with collecting the best design elements of each prototype, followed by iterative refinement.
- Plan with at least four development stages according to the activities needs analysis, automated testing, expert testing, and user testing.
- Let one development cycle contain the steps requirements formulation, implementation, integration, and evaluation.
- Do as much (semi-)automated testing as possible before evaluations that involve humans as this is a cheap process, but keep in mind that the majority of usability problems are found in user trials [21, 22].
- Let experts wipe out the most obvious usability and accessibility flaws before involving user groups.
- Prefer testing with people with disabilities over people without disabilites.
- Let at least the following subgroups be represented in each user evaluation: the sensor impaired (at least sight and hearing), the cognition impaired (e.g., dyslexia, etc.), the motor impaired (e.g., cerebral palsy), individuals with a combination of impairments (e.g., elderly), and individuals with low IT skills.



Name	Work	
Development of service XY	**72d**	
Kick-off	**3d**	
Introduction, planning, coordination	3d	NN1, NN2, NN3
Implementation	**62d**	
1. Iteration	**29d**	
Needs analysis and brainstorming (experts)	2d	NN1, NN3
3-4 Design alternatives as paper prototypes	4d	NN1, NN2, NN3
Design ranking (focus group)	2d	NN1 [50], NN3
Merging of best design elements to Working Draft	2d	NN1, NN2
Working Draft requirement spec, guidelines, etc.	4d	NN1 [70]
Implementation, integration	15d	NN1 [70], NN2 [50]
Pre-Alpha		NN2
2. Iteration	**6d**	
Evaluation of Pre-Alpha (semi-/automatic testing)	2d	NN1, NN3
Revision of Pre-Alpha requirements spec	1d	NN1 [50]
Implementation, integration	3d	NN2 [70]
Alpha		NN2
3. Iteration	**9d**	
Evaluation of Alpha (personas/expert testing)	2d	NN1, NN3
Revision of Alpha requirement spec	2d	NN1 [50]
Implementation, integration	5d	NN1 [50], NN2 [70]
Beta		NN2
4. Iteration	**14d**	
Evaluation of Beta (user trial)	2d	NN1, NN3
Revision of Beta requirement spec	2d	NN1 [70]
Implementation, integration	10d	NN1 [50], NN2 [70]
Release Candidate (RC)		
5. Iteration	**4d**	
Evaluation of RC (ekspert testing)	2d	NN1 [50], NN3
Implementation (bug fixes)	2d	NN1 [50], NN2 [50]
Golden		NN2
Documentation	**7d**	
Evaluation of Golden (user trial)	2d	NN1 [50], NN3
Documentation	4d	NN1 [70]
Delivery, training	1d	NN1

Fig. 1. Gantt diagram template for planning of user centered design activities. Activity durations are only set for illustration purposes.

- A subgroup should consist of five (three if more than two subgroups) to eight users; a minimum of five to cover 85% of usability issues, and a maximum of eight to avoid inefficient testing [37]. Bigger user groups should be split up into several iterations with smaller user groups; e.g., three iterations á five users instead of one iteration with 15 users [37].
- Design and test the user interface first, then complement it by the functionality of the underlying application and system.
- Make user trials as much hands-on/realistic as possible; e.g., let impaired users test with their own equipment and in their own context in the field, and let them use assistive technology.

5 Conclusion

We have discussed how universal-design aspects can be included in the software projects to develop inclusive ICT solutions, based on recommendations from relevant literature and on our own experience with small to medium-scale development projects. We also presented a list of best-practice measures to be applied to projects with deliverables which have to meet universal-design requirements and want to do so in an efficient manner, and a workflow template which can be used as an example.

By sorting out possible pit-falls and summarizing recommendations, this work anticipates to aid projects in making the right decision regarding workflow, activities, activity order, etc., with the overall goal to make any ICT solutions as accessible and usable as possible, and to reach this goal in an efficient way. It is suggested, though, that the recommendations given here be not applied without thought, but should be considered with regard to the respective situation particular for each development project.

References

1. Alliance, A.: Manifesto for agile software development (2001) (retrieved February 25, 2013)
2. Norman, D.: The Design of Everyday Things. Basic Books, New York (1988)
3. International Standardization Organization: Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. ISO 9241-210:2010 (2010)
4. Fox, D., Sillito, J., Maurer, F.: Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In: AGILE 2008 Conference, pp. 63–72. IEEE (2008)
5. Singh, M.: U-SCRUM: An agile methodology for promoting usability. In: AGILE 2008 Conference, pp. 555–560. IEEE (2008)
6. Hodgetts, P.: Experiences integrating sophisticated user experience design practices into agile processes. In: Proceedings of the Agile Conference, pp. 235–242. IEEE (2005)
7. Chamberlain, S., Sharp, H., Maiden, N.: Towards a framework for integrating agile development and user-centred design. In: Abrahamsson, P., Marchesi, M., Succi, G. (eds.) XP 2006. LNCS, vol. 4044, pp. 143–153. Springer, Heidelberg (2006)
8. Hussain, Z., Slany, W., Holzinger, A.: Current state of agile user-centered design: A survey. HCI and Usability for e-Inclusion, 416–427 (2009)
9. Sy, D.: Adapting usability investigations for agile user-centered design. Journal of Usability Studies 2(3), 112–132 (2007)
10. Vredenburg, K., Mao, J.Y., Smith, P.W., Carey, T.: A survey of user-centered design practice. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing our World, Changing Ourselves, pp. 471–478. ACM (2002)
11. Bowles, C.: Looking beyond user-centered design. Online (2013) (retrieved February 25, 2013)
12. Røssvoll, T.H.: The European MobileSage project – situated adaptive guidance for the mobile elderly: Overview, status, and preliminary results. In: Sixth International Conference on Advances in Computer-Human Interactions (ACHI), IARIA (2013)

13. Center for Universal Design, North-Carolina State University: The principles of universal design (April 1997), http://www.ncsu.edu/project/design-projects/udi/center-for-universal-design/the-principles-of-universal-design/ (retrieved February 28, 2013)
14. Aslaksen, F., Bergh, S., Bringa, O.R., Heggem, E.K.: Universal design: Planning and design for all (1997)
15. Vanderheiden, G.: Fundamental principles and priority setting for universal usability. In: Proceedings on the 2000 Conference on Universal Usability, pp. 32–37. ACM (2000)
16. Keates, S., Clarkson, P.: Countering design exclusion: bridging the gap between usability and accessibility. Universal Access in the Information Society 2(3), 215–225 (2003)
17. Stephanidis, C., Salvendy, G.: Toward an information society for all: An international research and development agenda. International Journal of Human-Computer Interaction 10(2), 107–134 (1998)
18. Stephanidis, C.: Designing for all in the information society: Challenges towards universal access in the information age. Informe técnico ERCIM ICS Report, Institute of Computer Science, Heraklion, Grecia (1999)
19. Newell, A.F., Gregor, P., Morgan, M., Pullin, G., Macaulay, C.: User-sensitive inclusive design. Universal Access in the Information Society 10(3), 235–243 (2011)
20. Wobbrock, J.O., Kane, S.K., Gajos, K.Z., Harada, S., Froehlich, J.: Ability-based design: Concept, principles and examples. ACM Transactions on Accessible Computing (TACCESS) 3(3), 9 (2011)
21. Rømen, D., Svanæs, D.: Validating wcag versions 1.0 and 2.0 through usability testing with disabled users. Universal Access in the Information Society, 1–11 (2011)
22. Power, C., Freire, A., Petrie, H., Swallow, D.: Guidelines are only half of the story: accessibility problems encountered by blind users on the web. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, pp. 433–442. ACM (2012)
23. Paddison, C., Englefield, P.: Applying heuristics to perform a rigorous accessibility inspection in a commercial context. In: ACM SIGCAPH Computers and the Physically Handicapped. Number 73-74, pp. 126–133. ACM (2003)
24. Petrie, H., Kheir, O.: The relationship between accessibility and usability of websites. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, vol. 28, pp. 397–406 (2007)
25. Zimmermann, G., Vanderheiden, G.: Accessible design and testing in the application development process: considerations for an integrated approach. Universal Access in the Information Society 7(1), 117–128 (2008)
26. Nischelwitzer, A., Pintoffl, K., Loss, C., Holzinger, A.: Design and development of a mobile medical application for the management of chronic diseases: methods of improved data input for older people. In: Holzinger, A. (ed.) USAB 2007. LNCS, vol. 4799, pp. 119–132. Springer, Heidelberg (2007)
27. Fuglerud, K., Dale, O.: Secure and inclusive authentication with a talking mobile one-time-password client. IEEE Security & Privacy 9(2), 27–34 (2011)
28. Fuglerud, K.S., Røssvoll, T.H.: An evaluation of web-based voting usability and accessibility. Universal Access in the Information Society, 1–15 (2011)
29. Fuglerud, K.S.: ICT services for every citizen: The challenge of gaps in user knowledge. In: Stephanidis, C. (ed.) Universal Access in HCI, Part I, HCII 2009. LNCS, vol. 5614, pp. 38–47. Springer, Heidelberg (2009)

30. Gregor, P., Newell, A.F., Zajicek, M.: Designing for dynamic diversity: interfaces for older people. In: Proceedings of the Fifth International ACM Conference on Assistive Technologies, pp. 151–156. ACM (2002)
31. Halbach, T.: A framework for serving inclusive web forms to disabled and elderly individuals. In: Proceedings of 2nd International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies, and Services, IARIA. IEEE Computer Society, Porto, Portugal (2009)
32. Hellman, R., Halbach, T., Nårstad, A.L.: Log in if you can – the Brønnøysund case. In: Proceedings of Verdiskapning i IKT (VERDIKT), Bergen (Norway), Norwegian Research Council (October 2008)
33. Røssvoll, T.H., Solheim, I.: Design of cognitively accessible web pages. International Journal on Advances in Intelligent Systems 3(3-4), 303–312 (2010) (last accessed April 4, 2011)
34. Røssvoll, T.H., Fritsch, L.: Reducing the user burden of identity management: A prototype based case study for a social-media payment application. In: Sixth International Conference on Advances in Computer-Human Interactions, ACHI (2013)
35. Niitamo, V.P., Kulkki, S., Eriksson, M., Hribernik, K.A.: State-of-the-art and good practice in the field of living labs. In: Proceedings of the 12th International Conference on Concurrent Enterprising: Innovative Products and Services through Collaborative Networks, pp. 26–28. Milan, Italy (2006)
36. Dow, S.P., Fortuna, J., Schwartz, D., Altringer, B., Schwartz, D.L., Klemmer, S.R.: Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In: Design Thinking Research, pp. 47–70 (2012)
37. Nielsen, J.: Why you only need to test with 5 users, 2000. Jakob Nielsen's Alertbox (2009), www.useit.com/alertbox/20000319.html