Active Involvement of Software Developers in Usability Engineering: Two Small-Scale Case Studies

Nis Bornoe(✉) and Jan Stage

Aalborg University, Aalborg, Denmark
{nis, jans}@cs.aau.dk

Abstract. The essence of usability evaluations is to produce feedback that supports the downstream utility so the interaction design can be improved and problems can be fixed. In practice, software development organizations experience several obstacles for conducting usability engineering. One suggested approach is to train and involve developers in all phases of usability activities from evaluations, to problem reporting, and making redesign proposals. Only limited work has previously investigated the impact of actively involving developers in usability engineering. In this paper, we present two small-scale case studies in which we investigate the developers' experience of conducting usability evaluations and participating in a redesign workshop. In both case studies developers actively engaged in both activities. Per the developers, this approach supported problem understanding, severity ratings, and problem fixing. At the organizational level, we found that the attitude towards and understanding of the role of usability engineering improved.

Keywords: Usability engineering · Redesign · Case study

1 Introduction

Usability evaluations are widely accepted and used during iterative software development for measuring usability and identifying flaws in the interaction design of IT systems [7]. One form of evaluation is the formative approach often conducted with a think-aloud evaluation [27]. Such evaluations are used to get feedback about users' behavior, and the flows, concepts, and designs used. Evaluations provide new and more detailed insights about the state of a given interaction design not already known to the developers [16]. Insights that can be used to improve and develop an application [32]. Several obstacles to deploying usability engineering into organizations have been found. This includes resource requirements, recruiting participants [5, 23, 24], lack of knowledge and competencies [1, 33], establishing credibility [9, 13, 22], and developers neglecting usability perspectives [1, 22]. On top, integration with software engineering processes is challenging [29], and organizational factors have an influence [13, 20], for example, often a decision about who has the responsibility of usability is missing [8]. In a study Vukelja and colleagues found that developers commonly develop user interfaces by themselves without the involvement of designers, they lack

R. Bernhaupt et al. (Eds.): INTERACT 2017, Part IV, LNCS 10516, pp. 159–168, 2017.
DOI: 10.1007/978-3-319-68059-0_10

relevant knowledge, do not effectively include users in the design process, and if they conduct an evaluation, it rarely has a substantial impact [33]. In addition, problem priority and severity assessment [14], fixing problems, and reevaluating systems are also commonly overlooked challenging activities [22, 31, 32]. Because fixing usability problems is an essential [14, 22, 35] non-trivial complex part of usability engineering [31, 32], concentrated focus and planning are required [12, 31].

To overcome some of these obstacles it has been suggested to involve and train the developers in usability engineering because this can increase interest and support for usability engineering [18, 21]. This approach is reported to be relevant for both organizations without usability competences and the ability to hire consultants [5, 11], and to improve cross-disciplinary collaboration in organizations employing both designers and developers [21, 26]. A study report that having developers observe evaluations resulted in increased empathy towards the users, and a comprehensive understanding of the severity and context of problems [15]. A study about involving developers in field studies report increased knowledge of the context of use, and a better understanding of user perspectives [11]. A study about training novices in think-aloud evaluations concludes that conducting the evaluation and creating user tasks went well, but the identification and description of usability problems were flawed in comparison to experts [30]. Involving developers in the process of creating redesign suggestions have also shown to be beneficial because they possess domain knowledge [12].

It has been pointed out that insights about the impact of usability engineering in real world projects are needed [22, 32] along with focusing research on concepts that can be implemented into practical settings [11, 17, 25]. We here present two small-scale case studies building upon previous work focused on providing developers with basic skills in usability engineering [3, 4, 6].

Our research question is: "*What are the experiences of actively involving the developers in usability engineering?*"

2 Method

We designed an interpretive case study and collected qualitative data. We chose this research design because software development projects are complex and dynamic and therefore challenging to study in isolation. A case study is a relevant research approach when investigating phenomena in the context of practical circumstances of real-world development projects where it is difficult to maintain fixed parameters, and the data collection process can be unpredictable [28].

Walsham points out that making generalizations based on case studies has been widely debated, but argue that the description of phenomena grounded in interpretation of empirical data can turn out to be beneficial for other organizations or contexts. For example, case studies can support generalizing the generation of theory or drawing of specific implications. He emphasizes that the outcome of case studies should be considered tendencies rather that a precise forecasting [34].

In the following, we will present the two case organizations, outline the activities they were involved in, and describe the data collection and analysis.

2.1 Cases

Both case organizations are relatively small. They employ about 20 developers each, and follow the agile development approach Scrum. They had no competencies and practical experience with conducting systematic usability engineering and interaction design. A few developers had limited knowledge about human-computer interaction and interaction design gained during their studies. Both organizations participated in similar usability evaluation training [6] and redesign activities [3, 4]. The purpose was to introduce usability engineering into the organizations and provide the development teams with basic competencies applicable in their day-to-day development practices. The idea was to provide a starting point from where the organizations would be able to evolve their usability engineering competencies further and conduct the activities on their own or request facilitation when needed. Below we outline the two activities.

- **Usability evaluation training.** The two organizations received training following the barefoot usability evaluation learning approach focused on providing elemental evaluation skills [6]. During a mini course, lasting about 30 h, they were taught how to conduct a think-allowed evaluation (TA), video analysis [27], and Instant Data Analysis (IDA), an approach used to analyze data from TA evaluations rapidly [19]. The usability mini course consisted of a mix of lectures and exercises, for example, they were asked to plan, conduct, and analyze an evaluation on their own under the supervision of usability specialists [6].
- **Redesign workshop participation.** The developers participated in redesign workshops facilitated by usability specialists [3, 4]. By facilitating problem fixing through a collaborative workshop actively involving the developers, the aim was to unite usability and domain knowledge, and by working in small groups, the developers can produce several alternative redesign suggestions. The workshop was not only about fixing the identified problems but also progressing the design. Firstly, they received a basic introduction to interaction design principles (only C2). Secondly, they discussed the problem list and picked a subset of problems. Thirdly, groups of three developers participated in focused discussions internally, and together with a usability specialist to outline redesign suggestions. Finally, in plenum, the groups presented and discussed their redesign suggestions.

Case organization 1 (C1). They develop self-service solutions and administrative forms for the public sector. This includes the frontend system for the users, and backend system for the administrative staff. Initially, they produced paper-based forms and had a portfolio of about 300. Interactive PDF documents replaced these forms, and finally, the interactive PDF documents were transformed into web-based services available through a self-service citizen portal. As it became a requirement for citizens to use the self-service solutions, usability became part of the system specifications. At this point the developers designed interaction designs haphazardly.

- **The system:** The system is an interactive PDF document used by citizens and companies when applying for a permit or submitting a notification required for construction of new buildings or certain types of renovations. Essentially the system is an interactive version of a paper-based form. This application was chosen because it was relevant to many citizens, and reasonably complicated to fill out.

- **Usability evaluation:** Four usability specialists conducted a TA usability evaluation with 10 participants. The participants had previous experience with public citizen services, but no experience with this system. The developers observed the evaluation. The result was a problem list consisting of 75 usability problems [4].
- **Redesign workshop:** Divided into three groups five developers including two programmers, two project managers, and a head of development participated in the redesign workshop along with four usability specialists. 25 out of the 75 problems were selected to make the workshop more focused and manageable. The outcome of the workshop was three suggestions for redesign suggestions [4].

Case organization 2 (C2). They develop audio equipment for the music industry. One product line is guitar and bass pedals, which are effects units used for altering audio source sounds. Recent years, they have increased the production of software extending the functionality of the pedals. The development team initially decided to increase the focus on interaction design and usability because of two main reasons: (1) users reporting difficulties to understand and use the advanced options in the interactive user interface, and (2) many decisions about interaction design were often based on the "gut feeling" of individuals. Instead, they wanted a methodical approach as the basis for making informed decisions. They wanted more of the team to get knowledge and understanding of the importance of having a methodical approach. In addition, they wanted to acquire or modify approaches practical in terms of their scrum-rhythm.

- **The system:** The organization has developed an application used to customize the settings of the guitar effect pedals they produce.
- **Usability evaluation:** The developers conducted a TA evaluation with four hobby musicians as participants. The evaluation resulted in a list of 19 usability problems. For each problem, the following was given: a short description, a severity rating in the form: minor, moderate and severe, a list of all evaluators identifying the problem, a short redesign suggestion, a complexity rating (1–8) decided by two programmers, and a business value rating (1–8) decided by two product managers. This rating was used to determinate which problems to prioritize and focus on [3].
- **Redesign workshop:** Divided into two groups six developers including two software developers, one hardware developer, two product managers, and one software quality assurance manager participated in the redesign workshop along with two usability specialists. The conclusion of the redesign workshop is that the developers participating were able to reconsider the existing design based on a top-down approach constructively and outlined new ideas for the use flow [3].

2.2 Data Collection and Analysis

We conducted semi-structured interviews because this allowed dynamic conversations, covered our set of questions, and made the answers comparable. From C1 we conducted an interview with the head of development (HoD). From C2 we conducted an interview with, a developer, and a product manager (PM). All three had participated in both the usability training and the redesign workshop. Additionally, we had some observation notes, and collected archival data in the form of usability problem lists, presentations, and sketches. For the interpretive analysis of the qualitative data, we followed the principles suggested by Elliott and colleagues [10].

3 Findings

We here present the findings divided into the impact of involvement in the usability evaluation, the redesign workshop, and what impact this had on the organization.

3.1 Impact of Involvement and Training in Usability Evaluations

Problem discovery and insights. The developers' involvement in the evaluations provided both insights about problems, but also underlined users' perception and actual use of the system. *"…the developers could see that there really were some major problems and could see people desperately get stuck with a solution they had developed."* (HoD, C1) In addition to getting aware of new problems, they found it useful to get confirmation or disconfirmation of problems they were already suspecting. They got a more detailed understanding of the problems, and their fuzzy ideas were concretized and extended. *"Maybe we had the feeling that it could be done better, but it was not made completely clear and formulated explicitly."* (PM, C2) For example, C2 found that the flow of operations on a particular screen was not in line with the flow of operations found logic by the users, and how the users wanted to interact with the application. Getting direct insights into this design flaw was by the PM in C2 characterized as *"…a big eye opener…"*, and the developer added: *"We had not thought of the flow as a major problem. It is not certain that the order of elements would be different had we not conducted a user test."* (Developer, C2) In summary, these direct insights increased an understanding of the users, as also reported by an earlier study [15]. This understanding from interacting with, and seeing users in action made fuzzy ideas of problems specific, more detailed and inspired and motivated the developers to make changes. One challenge was the thoroughness of problem identification and severity ratings [6].

Prioritizing and rating problems. Being part of the usability evaluation supported the prioritization process, and forced the organizations to actively reconsider their current strategy. *"When you are used to making systems in a certain way, you just keep doing the same."* (HoD, C1) During the compilation of the list, both organizations provided a classic severity rating in the form 'minor,' 'moderate,' and 'severe.' A study of the training of C1 found that the developers had success conducting and creating user tasks, but the thoroughness of the problem identification and severity rating was less successful [6]. To partly overcome this obstacle, C2 experimented with two additional ratings. The interviewed programmer and a colleague would give a complexity rating (1–8). This rating is the estimated technical complexity of fixing the problem. The interviewed project manager and colleague would give a business value rating (1–8). This rating is the estimated importance related to the functionality of the application.

Both are also related to the resource requirement estimation for fixing a given problem. The three ratings (severity, complexity, and business value) were then used to decide which problems to prioritize. Through this prioritizing process, the development team could understand and analyze the problems from more angles. This also served to make the fixing of usability problems more specific and goal oriented. The developer noted that: *"The problem list had much greater influence than was intended when we made it."* (Developer, C2) In summary, being part of all steps of the evaluation made the identified problems relevant and creditable to the developers. Instead of simply adding problems to the backlog, there were some clear thoughts behind what problems to prioritize.

3.2 Impact of Participation in the Redesign Workshop

Intermediate design phase. The redesign workshop acted as an intermediate phase with a dedicated and concentrated focus outlining specific problems and what could be done to solve the problems. For both organizations, the redesign workshop was a major brainstorming session. As also reported in a recent study [12], they could use their extensive domain knowledge and produce several redesign suggestions or ideas with a short time span, while receiving guidance from a usability specialist. They did not generate implementable designs but used this session to dig into the specific problems. The division into groups and the afterward discussion provided insights from different angles and redesign proposals. Inspiration was brought back home for further refinement *"…we had something concrete to bring back home."* (HoD, C1) *"…after we participated in the workshop we had a list of things you could change, things that could be interesting…"* (PM, C2).

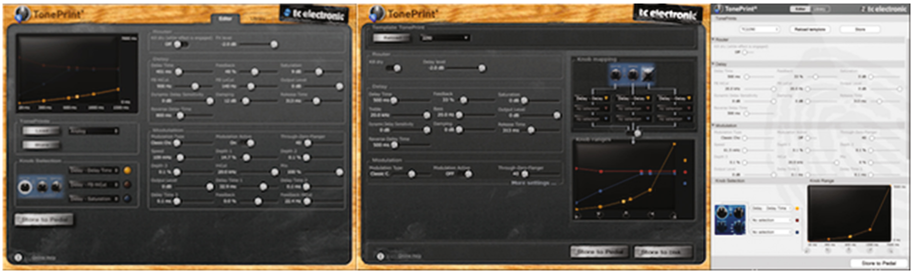

Fig. 1. The interface redesign by C2. From left: initial design, first revision, and final version.

In the immediate period following the redesign workshop, C1 engaged in an iterative design process and had discussions with the usability specialists and showcased prototypes to get feedback. C2 also wanted to make technical changes to the system and decided to merge two systems into one. During this process, they returned to the inspiration from the workshop. *"We had these things on a list that we looked over and prioritized in relation to what we thought was the most important and relative to what was comparatively affordable."* (PM, C2).

Of criticism, it was mentioned that a stricter frame for the part of the workshop concerned with actually making redesigns was wanted. For example, specific design exercises, and a better way of collecting and comparing the ideas generated.

Design changes. Both organizations redesigned and released new systems. During the redesign workshop, C1 decided on a wizard approach as the overall design pattern. Over a couple of months, they redesigned and implemented a new solution. They changed the form into a wizard approach with questions for each step. The essence of the approach was *"…to reformulate [the form] into something people can decide on."* (HoD, C1) This approach was used to derive the information needed and to guide the user through the application instead of requiring the users to figure out which fields are relevant and what and how to fill in the requested information. When evaluating the redesigned application, it was found that the handling of an application by a caseworker had decreased from an average of 53 min to 18½ min [2]. This indicates that a lot fewer applications with errors were submitted. The intention was to develop some principles based on this case that could be used in the other solutions.

C2 released the second revision of their app two years later. During the redesign process, a couple of significant design changes were made. The flow of operations and order of options on a screen was found to be problematic, something they noticed during the evaluation. While this was not a specific usability problem, the development team decided to work on this problem during the redesign workshop. During the initial design of the application, they wanted to make the application 'flashy.' *"[This element] is a fine smart impressive graph, an eye catcher, therefore we had probably kept it as a prominent element [in the top]."* (Developer, C2) During the workshop, they instead created redesign proposals based on the insights from the evaluation and the basic interaction design principles introduced. Afterward, they further evolved these proposals into a specific deign and implemented it (see Fig. 1).

3.3 Impact on the Organization

The developer effect. There was skepticism about usability engineering and its importance among serval developers, an obstacle reported by several other studies [1, 22]. The original believe in C1 about improving the usability of the building permit form was that: *"It is the user that has to obtain knowledge about how to fill in these forms, that it is not our problem."* (HoD, C1) The focus shifted from purely focusing on technical aspects to also considering usability as a quality factor for good software development. *"It means a lot [when] you deliver a system and receives positive feedback about the solution."* (HoD, C1) C2 also acknowledged that they had not prioritized usability *"You tend to neglect such things."* (PM, C2) *"The user evaluation [of our application] highlighted how big the problems were. We would probably not have realized that if we did not make that evaluation."* (Developer, C2).

Prime mover. Because usability engineering was not the primary concern or skill of any developers involved in the project, it was essential to have a prime mover, a person acting as the primary driving force and spearhead. Someone needed to take responsibility, and promote, manage, and prioritize these activities. In C1 the head of development took this role, and in C2 a developer that have completed several university level courses, and an interest in usability took this role. *"…to have someone on the team that is the prime mover that this is one of the important things to make it cool, ensuring it's done right and that someone considers this. If it is me, then it will just be a priority in many priorities."* (PM, C2) C2 has since hired a dedicated UX specialist to act as the prime mover.

Towards integrating user-centered design. After participation, both organizations have on their own conducted evaluations of other products and experimented with user-centered activities. However, the activities in C1 later stopped because of a change in management, and several key people left the organization. Opposite, C2 has experimented with future technology workshops and contextual inquiry and design. This is still at an early stage, and they have yet to figure out exactly how to proceed and use the output of such activities. *"We held a workshop…[where]…we focused on what could make sense for users in terms of integrating [our products] with the Internet and social media."* (Developer, C2) In another project, they wanted early feedback from users. *"'Here's an app and pedal, they are connected in one way or another. Try to solve this problem or how will you do this?' We observed if they could figure it out or not. Based on that, we changed it along the way as there were some things they were having troubles with."* (PM, C2).

4 Conclusion

Based on two small-scale case studies, we have presented developers' experiences of engaging in an evaluation and redesign process. Both organizations had previously learned the basics of conducting a TA evaluation and how to organize a redesign workshop to come up with different redesign proposals. In response to the usability activities, they have afterward come up with proposed fixes for several identified usability problems and, perhaps more importantly, made some fundamental design changes. We learned and will advocate that by being a continuous part of the process, developers get a better understanding of the roots to underlying usability areas not necessarily explicit identified as specific usability problems. This knowledge is hard to get by only reading a usability report, but is valuable to make systems better and to change the mindset of an entire organization by explicating specific problems and showing specific design changes.

While both organizations are positive and believe they have gained new strategic knowledge, the caveat is that maintaining the skills require work by the involved parties. To avoid that the skills do not fade away but progress it is essential to have this acknowledged and prioritized by management as well as a prime mover taking the lead and acting as the spearhead. The two introduction activities only provide a basic entrance to usability engineering and do not make them experts [6]. This integration is a long-term commitment. The training and involvement only acted as a starting point from which the organizations need to evolve skills further.

References

1. Bak, J.O., et al.: Obstacles to usability evaluation in practice: a survey of software development organizations. In: Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges, pp. 23–32. ACM, New York (2008)
2. Billestrup, J., et al.: UX requirements to public systems for all: formalisation or innovation. In: INTERACT 2015 Adjunct Proceedings: 15th IFIP TC. 13 International Conference on Human-Computer Interaction, Bamberg, Germany, p. 2015407, 14–18 September 2015
3. Bornoe, N., et al.: Redesign workshop: involving software developers actively in usability engineering. In: Proceedings of the 8th Nordic Conference on Human-Computer Interaction Fun, Fast, Foundational - NordiCHI 2014, pp. 1113–1118. ACM, New York (2014)
4. Bruun, A., Jensen, J.J., Skov, M.B., Stage, J.: Active collaborative learning: supporting software developers in creating redesign proposals. In: Sauer, S., Bogdan, C., Forbrig, P., Bernhaupt, R., Winckler, M. (eds.) HCSE 2014. LNCS, vol. 8742, pp. 1–18. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44811-3_1
5. Bruun, A.: Training software developers in usability engineering: a literature review. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, pp. 82–91. ACM, New York (2010)
6. Bruun, A., Stage, J.: Barefoot usability evaluations. Behav. Inf. Technol. **33**(11), 1148–1167 (2014)
7. Bygstad, B., et al.: Software development methods and usability: perspectives from a survey in the software industry in Norway. Interact. Comput. **20**(3), 375–385 (2008)
8. Cajander, Å., Larusdottir, M., Gulliksen, J.: Existing but not explicit - the user perspective in scrum projects in practice. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) INTERACT 2013, Part III. LNCS, vol. 8119, pp. 762–779. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40477-1_52
9. Chilana, P.K., et al.: Understanding usability practices in complex domains. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2337–2346. ACM, New York (2010)
10. Elliott, R., et al.: Descriptive and interpretive approaches to qualitative research. Handb. Res. Methods Clin. Heal. Psychol. **1**, 147–159 (2005)
11. Eriksson, E., Cajander, Å., Gulliksen, J.: Hello world! – experiencing usability methods without usability expertise. In: Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (eds.) INTERACT 2009. LNCS, vol. 5727, pp. 550–565. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03658-3_60
12. Garnik, I., et al.: Creative sprints: an unplanned broad agile evaluation and redesign process. In: Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, pp. 1125–1130. ACM, New York (2014)
13. Gulliksen, J., et al.: Usability professionals—current practices and future development. Interact. Comput. **18**(4), 568–600 (2006)
14. Hertzum, M.: Problem prioritization in usability evaluation: from severity assessments toward impact on design. Int. J. Hum. Comput. Interact. **21**(2), 125–146 (2006)
15. Høegh, R.T., et al.: The impact of usability reports and user test observations on developers' understanding of usability data: an exploratory study. Int. J. Hum. Comput. Interact. **21**(2), 173–196 (2006)
16. Høegh, R.T., Jensen, J.J.: A case study of three software projects: can software developers anticipate the usability problems in their software? Behav. Inf. Technol. **27**(4), 307–312 (2008)
17. Juristo, N., et al.: Analysing the impact of usability on software design. J. Syst. Softw. **80**(9), 1506–1516 (2007)

18. Karat, J., Dayton, T.: Practical education for improving software usability. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 162–169. ACM Press/Addison-Wesley Publishing Co., New York (1995)
19. Kjeldskov, J., et al.: Instant data analysis: conducting usability evaluations in a day. In: Proceedings of the Third Nordic Conference on Human-Computer Interaction, pp. 233–240. ACM, New York (2004)
20. Kuusinen, K., Mikkonen, T., Pakarinen, S.: Agile user experience development in a large software organization: good expertise but limited impact. In: Winckler, M., Forbrig, P., Bernhaupt, R. (eds.) HCSE 2012. LNCS, vol. 7623, pp. 94–111. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34347-6_6
21. Latzina, M., Rummel, B.: Soft (ware) skills in context: corporate usability training aiming at cross-disciplinary collaboration. In: Proceedings of the 16th Conference on Software Engineering Education and Training (CSEE&T 2003), pp. 52–57 (2003)
22. Law, E.L.-C.: Evaluating the downstream utility of user tests and examining the developer effect: a case study. Int. J. Hum. Comput. Interact. **21**(2), 147–172 (2006)
23. Lizano, F., et al.: Is usability evaluation important: the perspective of novice software developers. In: The 27th International BCS Human Computer Interaction Conference (HCI 2013) (2013)
24. Monahan, K., et al.: An investigation into the use of field methods in the design and evaluation of interactive systems. In: Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction, vol. 1, pp. 99–108. British Computer Society, Swinton (2008)
25. Nørgaard, M., Hornbæk, K.: What do usability evaluators do in practice?: an explorative study of think-aloud testing. In: Proceedings of the 6th Conference on Designing Interactive Systems, pp. 209–218. ACM, New York (2006)
26. Øvad, T., et al.: Teaching software developers to perform UX tasks. In: Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction on - OzCHI 2015, pp. 397–406. ACM (2015)
27. Rubin, J., Chisnell, D.: Handbook of Usability Testing [Electronic Resource]: How to Plan, Design, and Conduct Effective Tests, 2nd edn. John Wiley & Sons (2008)
28. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131 (2009)
29. Salah, D., et al.: A systematic literature review for agile development processes and user centred design integration. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, pp. 5:1–5:10. ACM, New York (2014)
30. Skov, M.B., Stage, J.: Training software developers and designers to conduct usability evaluations. Behav. Inf. Technol. **31**(4), 425–435 (2012)
31. Smith, A., Dunckley, L.: Prototype evaluation and redesign: structuring the design space through contextual techniques. Interact. Comput. **14**(6), 821–843 (2002)
32. Uldall-Espersen, T., et al.: Tracing impact in a usability improvement process. Interact. Comput. **20**(1), 48–63 (2008)
33. Vukelja, L., Müller, L., Opwis, K.: Are engineers condemned to design? A survey on software engineering and UI design in Switzerland. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007, Part II. LNCS, vol. 4663, pp. 555–568. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74800-7_50
34. Walsham, G.: Interpretive case studies in IS research: nature and method. Eur. J. Inf. Syst. **4** (2), 74–81 (1995)
35. Wixon, D.: Evaluating usability methods: why the current literature fails the practitioner. Interactions **10**(4), 28–34 (2003)