

Focal Points for a More User-Centred Agile Development

Silvia Bordin^(✉) and Antonella De Angeli

Department of Information Engineering and Computer Science, University of Trento,
via Sommarive 9, 38123 Trento, Italy
{bordin, antonella.deangeli}@disi.unitn.it

Abstract. The integration of user-centred design and Agile development is becoming increasingly common in companies and appears promising. However, it may also present some critical points, or communication breakdowns, such as a variable interpretation of user involvement, a mismatch in the value of documentation, and a misalignment in iterations. We refine these themes, emerging from both literature and previous fieldwork, by analysing a case study performed in an IT company that adopts both software engineering approaches, and we further extend the framework with a new theme related to task ownership. We argue that communication breakdowns can become focal points to drive action and decision for establishing an organisational context acknowledging the value of user involvement: to this end, we suggest the adoption of design thinking and the active engagement of the customer in embracing its values.

Keywords: Communication breakdowns · Organisational culture · Case study

1 Introduction

In recent years we have witnessed a growing interest in the integration of Agile methodologies with user-centred design (UCD), in order to achieve a more holistic software engineering approach. In fact, UCD and Agile show some complementary aspects: on the one hand, UCD does not address how to implement the software, while Agile provides large flexibility in accommodating changing requirements; on the other hand, Agile does not directly address user experience (UX) aspects, although valuing customer involvement in the development process.

However, even though the integration of UCD and Agile appears promising, it also presents some issues and no fully satisfactory approach to it has been found yet. In particular, three communication breakdowns [4] hampering such integration have been identified [5], namely a variable interpretation of user involvement, a mismatch in the value of documentation, and a misalignment in iteration phases. In this paper, we refine this framework by discussing a new case study looking at the practices of a software and interaction design company. To support our analysis, we define the main actors involved and how they are mutually linked in a communication network, comparing the latter with the one resulting from the case study presented in [5]. Despite the differences in the two working contexts, the three themes manifest anyway and an additional point, related to task ownership, emerges. We conclude by discussing how these

communication breakdowns can become focal points to support action and decision in companies adopting UCD and Agile; moreover, we argue that possible solutions to these issues need to be backed by a supportive organisational culture that recognises the value of user contribution and actively endorses it with the customer.

2 Related Work

User-centred design (UCD) is an umbrella term used to denote a set of techniques, methods, procedures that places the user at the centre of an iterative design process [25]. The benefits of involving users in systems design are widely acknowledged [1, 14, 16, 18]: they include improved quality and acceptance of the system [11], and cost saving, since unnecessary features or critical usability issues are spotted early in the development process [23]. In recent years, there have been several attempts at integrating UCD with Agile software development, as witnessed for instance by the literature reviews in [15, 26]. Despite the large common ground that the two approaches share, there are at least three themes on which their perspectives diverge [5]: we frame these themes by drawing on the concept of communication breakdown, that is a “disruption that occurs when previously successful work practices fail, or changes in the work situation (new work-group, new technology, policy, etc.) nullify specific work practices or routines of the organizational actors and there are no ready-at-hand recovery strategies” [4]. Although originally discussed with respect to global software development, we believe that this concept can support a reflection on the synthesis of different software engineering approaches: we argue, in fact, that it refers to issues occurring at “work practice level” that are due to an “underdeveloped shared context of meaning” [4], which could also be interpreted as the incomplete establishment of a common ground [10] between designers and developers of the same company.

The three communication breakdowns in the integration of UCD and Agile were formalised during a field study carried out within the Smart Campus project [5], where UCD and Scrum were integrated in a process of mobile application development for a community of users, namely students of the University of Trento campus. The goal of this R&D project was to create an ecosystem fostering students’ active participation in the design and development of mobile services for their own campus [12]; more details about the aims and results of the project can be found in [6, 12, 34]. In the following, we will illustrate the three communication breakdowns identified by drawing on the literature review that supported the findings of the Smart Campus field study.

User Involvement. In UCD, user involvement can range from informative, to consultative, to participative [11]. In Agile instead, the emphasis is rather put on the customer [1], who acts as a representative of users, but may or may not have direct and regular contact with them [27, 28], to the point that some authors question the extent of such representativeness [30] and others recommend that the customer role is supported by members of the project team [9].

Documentation. Both UCD and Agile encourage frequent communication among team members; however, there can be issues in the communication between designers

and developers [1] and in the role of documentation in this respect. In fact, UCD suggests the use of several artefacts such as personas and prototypes to record requirements and design rationales [28], while Agile promotes face-to-face conversation as the most effective means of communication in its fundamental principles [3], to the point of incorporating the customer in the development team.

Synchronisation of Iterations. There are different schools of thought about whether UCD and Agile should be merged into a unified software engineering process, leveraging on their common practices [19, 35, 37], or should just proceed in parallel [20, 24, 33].

3 H-umus

We will now discuss a field study performed in H-umus, presented in their website as a “software and interaction design company”. Born in 2007 in one of the most well known Italian venture incubators, H-umus designs and develops mobile sales tools for the fashion industry and now belongs to a large Italian software and services business. The personnel include a CEO, a CTO, four project managers (two of whom are also interaction designers), and five developers. The company adopts a customised version of Scrum for the development and follows a loose interaction design approach. At present, H-umus offers two main products to an established customer portfolio: a B2B merchandising platform and a time and expenses accounting tool. The company also follows some ad-hoc projects for more occasional customers: we consider here the development of a mobile tool for a leading fashion brand that we will call FashionX.

3.1 Field Study Methodology

The field study was carried out by one of the authors and is summarised in Table 1: it consisted of 20 h of observation of working practices, semi-structured interviews, attendance to meetings. Furthermore, artefacts used to support work were examined, while interviews were transcribed and thematically analysed [29].

Table 1. Summary of field study activities performed at H-umus.

Day	Activity	Duration
October 26 th , 2015	Attendance of sprint planning meeting; interviews with the CEO, a project manager, a designer and a developer	7 h
November 20 th , 2015	Interviews with both designers and the CTO	6 h
December 14 th , 2015	Attendance of sprint planning meeting; interviews with two developers, a designer, and a project manager	7 h

3.2 Communication Network

This section will illustrate the actors involved in H-umus and how, possibly through some artefacts, they are connected in a network, as shown in Fig. 1. The dialogue with users is completely mediated by the customer, usually represented by the IT department of a large fashion business. The customer in turn communicates with H-umus through a project manager of this company, who is often also an interaction designer; such dialogue is supported by a series of artefacts such as requirements documents, prototypes, and cost or time estimates, which will be described more in detail in later paragraphs. The project manager is then usually the only point of contact between the inside and outside of H-umus: he collaborates with the management (i.e. the CEO) in the early stages of an approach to a new customer, with the CTO in the definition of the technical analysis, and with developers during the implementation. Internal communication is also supported by a range of artefacts. Finally, the owner group refers to the management for products developed on their behalf.

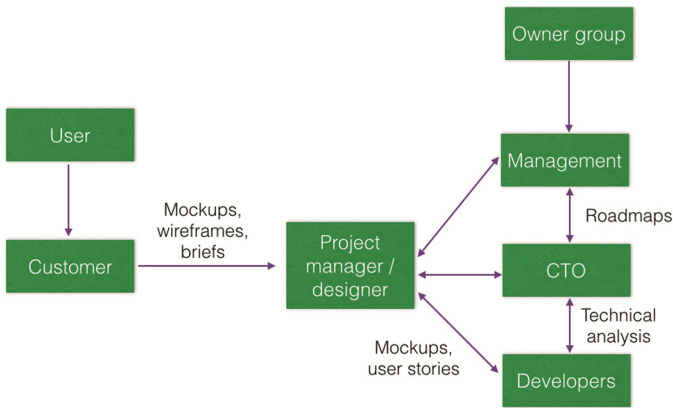


Fig. 1. Communication network in H-umus.

3.3 Artefacts

A variety of artefacts are used in H-umus to support communication, both internally and with the customer. In this paragraph, we will describe the most relevant ones.

Mockups and Wireframes. In the case of enhancements to already consolidated products, designers prepare high-fidelity mockups relying on the existing interface; in the case of software built from scratch instead, they prepare wireframes, representing interaction flows and layouts. Mockups and wireframes are then iteratively discussed with the customer: this allows to check that requirements have been correctly understood, to ensure that the customer is aware of project status and will not change his mind later, and to skip formal validation steps at the end of each sprint.

Briefs. Prototypes and requirements are integrated in documents called briefs, which crystallise the requirements; they are then iteratively revised with the customer to ensure that both parties share the same understanding of requirements and status of advancement.

Roadmaps. For each project, the relevant project manager keeps a chart showing the evolution of the product at a high level, including milestones to be delivered to the customer. This chart is often linked to other documents reporting, for instance, more extensive descriptions of functionalities or specifications of the customer's target platforms. Roadmaps are used internally, at management level: the CEO, the CTO and project managers refer to them to supervise the status of each project. However, if the customer requires so, roadmaps are also used to provide long-term visibility on the articulation of the project.

Technical Analysis. The CTO elaborates this document for each project: it includes finalised interface mockups, a description of the data flow and of the data structure, cost and time estimates, and a finer-grained breakdown of development tasks. The technical analysis serves two purposes: internally, it is a reference for developers to determine what to implement in the next sprints; externally and if needed, it can provide the customer with a detailed understanding of the implementation process.

3.4 Findings

In the following, we discuss the results of the interviews with the H-humus staff, categorising the narratives according to the three communication breakdowns constituting our framework. Citations in the next paragraphs will be attributed to interviewees as follows: *Dev* for developers; *Des* for designers; *PM* for project managers who are not designers; *Mgmt* for the CTO and the CEO.

User Involvement. The distinction between customers and users is very sharp and project managers usually communicate only with the customer, who can be represented by different employees at different stages of the same project. Especially when the customer is a large company, its most appropriate representative to liaise with can be difficult to identify and often changes over time:

Dev2: "The most difficult thing in communicating with the customer is understanding who you should be talking to."

In general, the customer representative is the IT department:

Mgmt2: "You would not believe how conservative IT departments can be. Whatever change may affect their working routine, it's a no-no."

There are, however, exceptions to this situation: for example, a few demos were arranged with business and sales representatives of FashionX, i.e. with a sample of final users, in order to collect feedback that could supplement the requirements provided by

the IT department of the company. Yet, this only happens occasionally: usually, and as shown in Fig. 1, the customer completely mediates user needs, requirements, and feedback. This causes some concern in the H-umus management:

Mgmt2: “Then it is difficult to determine how to handle the feedback we receive and how relevant it actually is with respect to the customer or with respect to the needs users may truly have. [...] Sometimes I wonder whom we should really satisfy. Is it the business department or the IT department? We usually speak only to the latter. I believe this causes a large drop in the value we deliver with our products.”

H-umus designers acknowledge that it would be desirable to apply a proper user-centred design methodology, involving real users in requirement gathering and interface evaluation. However, this is very hard to achieve in practice, because of two main reasons: first, the time for design is constrained; second, it is difficult to gain access to users. In fact, the customer is not always interested in being actively involved in the design of the commissioned product: sometimes H-umus may only be asked to prototype a new graphical interface for an existing software. The customer may even believe that users are not able to provide any sensible contribution:

Dev1: “I do not have any contact with users [...] Sometimes they are even described to me as being as dumb as an ox, so it is paramount to design products that are very easy to use, and I guess this is a major challenge for designers.”

Documentation. The staff has a small size and is co-located in the same open space: hence, most coordination occurs face to face or at most through instant messaging, both among developers and between developers and designers. This leads to a scarcity of documentation for internal use. However, in order to avoid knowledge gaps in case someone leaves the company, pair programming is adopted when a part of the code needs to be modified: the task is in fact assigned both to the developer who already worked on that code and to a “fresh” developer at the same time. In this way, in the long run everybody will have at least an overview of all the code produced. Working in pairs is also a common practice in the early stages of a new project, where a designer and a developer cooperate in order to shape the design space quickly and based on an understanding of what can be technically feasible.

PM1: “Everybody has an overview, but also a specific responsibility.”

Documentation is instead actively and carefully maintained to support the relationship with the customer. Despite the Agile principle [3] of “embracing change”, the management highlighted the need of making the customer responsible for his requirements and committed to them. The CTO and the project managers in fact insisted on their strong need to shield H-umus from sudden, important changes in customer requirements; being the company so small, this could cause a lot of work to be wasted and not paid, causing in turn potentially severe financial issues.

PM1: “H-umus is a small company. If the customer first says he wants a mobile app, and then after six months he comes and says that now he wants a standalone application... We cannot afford that. Unless the customer is paying for the extra time, of course.”

Des2: “We do not have much development capacity. It can become a big issue if I draw the mockup and then we have to go back and change fundamental parts of it.”

This protection is achieved by using several artefacts that are admittedly not typically Agile: documents such as requirements lists and technical analyses are shared with the customer, iteratively discussed and then signed off.

Mgmt1: “We make the customer sign the requirements document, so nobody can come up and say: “This is not what we agreed upon”. Whatever extra, we discuss it and it is billed on top.”

Des2: “Being able to tell the customer: “Look, this is what we suggested and you approved it” is something that can cover our back when we need to ask for more funding or when we just say that something is not feasible”.

The strong perception of documentation as having a purpose mainly in relation to the customer emerges very clearly also in relation to other themes:

Mgmt1: “I’ll show you the technical analysis we did for FashionX [...] Please write down in your notes that to me this is complete nonsense. The risk estimates and the planning poker and stuff... It is obvious that these numbers are meaningless. Yet the customer wants to have a long-term perspective on the project, so here it is.”

Synchronisation of Iterations. Given the small size of the company, designers and developers work together, so synchronisation is handled through constant, direct communication. Indeed, there is no separate process for design and for development: for instance, design tasks such as prototyping are listed as regular user stories in the Agile management tool in use:

Des1: “UX aspects are regarded as common functionalities.”

Despite a general awareness among the staff of the company transitioning towards a more design-oriented culture, the overall attitude appears to be still strongly technical. For instance, sprint meetings only involve developers:

Mgmt1: “We are born as a data-driven company [...] Sprint meetings are too technical; designers would waste time attending them.”

Furthermore, a different theme emerges, related to the recognition of designers’ expertise in a technically dominant environment. Several times designers referred to their competence in UX as being interpreted as common sense in the company:

Des2: “Why should the CEO’s opinion be more relevant than mine, if I designed the interface from the beginning? Sometimes [Des1] and I refer to it as a class conflict with the developers”

Des2: “Everybody feels entitled to comment on the design, just because each of us is a technology user, while nobody would comment on the code unless competent. So [developers] bring in their own use cases, but we are not developing, say, Instagram, which only has a couple of functionalities: it is totally different. Sometimes the comments are just “I don’t like it”. I can take it from the customer, if he pays for the extra time needed to rework the design, otherwise I’d expect some sounder feedback.”

The rest of the team perceives this issue as well, although in variable ways:

Dev1: “Interfaces are subjective [...] usability is subjective too: you need to design stuff that is comfortable for the user, more than functional. [Des1 and Des2] do a great job in my opinion in this respect.”

PM1: “The best way to work shouldn’t be to tell the designer how to do the things, but just what you need; unfortunately, the customer is often unable to articulate what he wants, and anyway we must give priority to the development to save time.”

Dev2: “We all give our opinion, but in the end it is the designer who decides.”

4 Discussion

Despite a positive attitude towards UCD, H-umus found objective difficulties in integrating it with Agile in practice. These difficulties were partially overlapping with the communication breakdowns identified in Smart Campus [5], although the working context of the latter was quite different from the H-umus one as illustrated by Fig. 2, which represents the main actors in Smart Campus and their communication network.

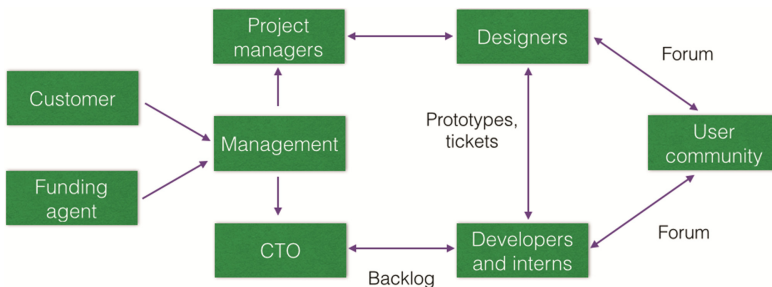


Fig. 2. Communication network in Smart Campus.

The analysis of the H-umus case study allowed us to refine our framework, broadening the scope of identified communication breakdowns as follows.

User Involvement. In Smart Campus, the customer and the user community were two clearly differentiated actors; most of the team had direct contact only with the users through a variety of communication channels such as a forum. However, the perception of user involvement appeared to be variable between designers and developers, denoting an underlying mismatch in the understanding of this concept: while designers struggled to promote a participative role of the user community, developers intended such role as informative or at most consultative instead [11]. In H-umus, the extent of user involvement remains problematic, although with a different flavour: the customer completely mediates the interaction with the user, so the role of the latter is practically less than informative [11]. Therefore, we can argue that the understanding of the extent of user involvement should be shared not only inside the company (among designers, developers, managers), but also outside, by the customer.

Documentation. In Smart Campus, documentation did not appear to have an intrinsic value as a communication tool for developers; however, it became increasingly relevant to keep the development team aligned when the latter became more distributed due to the introduction of interns working at variable times and often remotely. Yet, how to effectively support the need for a shared knowledge base remained an open point, particularly referring to design artefacts, although the team tried to adopt a variety of articulation platforms. In H-umus instead, the team is co-located: in this case, besides being a tool for tracing the history of the software and the rationale of related design and development choices, documentation can also have an instrumental function in balancing the power relationship with the customer, protecting the company against unsustainable changes in requirements.

Synchronisation of Iterations. The Smart Campus project was oriented towards a large and strong user community, whose feedback escalated quickly and was not mediated (for instance by a customer). This caused severe difficulties in synchronising the iterations of UCD and Agile: designers struggled to elaborate requirements and provide suggestions in a timely manner that could fit the development pace, while developers often took the initiative of fixing interfaces regardless of the overall UX vision. In general, designers resorted to several ad-hoc interventions, elaborated together with the developers requesting them. In H-umus instead, the team is co-located and quite small, so synchronisation can easily occur through face-to-face communication. Furthermore, the existence of signed documents prevents the customer from changing requirements with the same frequency witnessed in Smart Campus with the user community.

Task Ownership. An additional communication breakdown strongly emerged from the interviews conducted in H-umus. Several interviewees argued that, in order for an effective communication to occur, it is advisable that the whole team shares a common language. Additionally, our observations suggested that the team should also share a common understanding about who is responsible for each task, especially in the case of UX activities, and in particular for taking final decisions over it. This will help avoid situations in which a technically predominant environment interprets UX as mere “common sense”, which are not conducive to endorsing the added value that UX can provide to a product and which seem to reflect a long-lasting contrast between soft and

hard sciences. To this end, we point to the concept of boundary objects, i.e. mediating artefacts that allow knowledge sharing and promote collaboration since their interpretive flexibility facilitates “an overlap of meaning while preserving sufficient ambiguity” for different groups to read their own meanings [2]. The briefs used in H-umus can be considered as boundary objects in this sense, as they gather mockups from designers, technical specs from developers, and business requirements from the customer, and they act as a common reference point for monitoring the evolution of the product.

5 Conclusion

In this paper we have discussed four communication breakdowns that may affect the integration of user-centred design and Agile development and that emerged from an analysis of working practices in companies. Possible solutions can derive from discount usability techniques [e.g. 13, 22] or more recent research on automatic usability evaluation tools [e.g. 21, 31]. However, we remark that communication breakdowns are manifested at the work process level [4, 5]: hence, we suggest that their solution could be found in a supportive organisational environment [5, 8, 11, 17], whose fundamental importance is reiterated by the present study. As seen in H-umus, not even having designers play the role of project managers is enough to fully endorse the UCD component of the working process. To leverage the full potential of the integration of UCD and Agile, the management should actively counteract the so-called “developer mindset” [1, 14], i.e. an approach that is overly focused on technical aspects rather than on customer and user satisfaction, and commit to an explicit inclusion of UCD in company goals and financial allocation [36].

We claim that the four communication breakdowns discussed in this paper can become focal points to drive action and decision in companies, facilitating communication between designers and developers and supporting management in the construction of a favourable context. Our current research is addressing the development of specific guidelines concerning how to apply such focal points in practice through additional case studies. Nonetheless, and as already suggested in [5], we believe that design thinking [7] can be an appropriate methodology in this respect: grounded on a “human-centred design ethos”, it advocates a “designer’s sensibility” pervading the whole organisation, so that also technical personnel (be it part of the development or of the management) can be aware of the importance of meeting users’ needs with what is technologically feasible. Inspired by design thinking, the organisational culture is likely to empathise more with the user and to share the ownership of the UX vision among all members of the company: this is in turn also likely to address the task ownership theme introduced above.

However, the benefits of this internal culture may be limited if the customer does not share its same values, preventing access to users or completely mediating the communication with them. A direct contact with users can allow the company to deliver a product that, although requiring a possibly longer design period, will be more suited to the needs of people ultimately using it and will therefore bring more value to the customer for its money. Even after many years from [23], we still need to address the

“developer mindset” [1, 14] and persuade the customer and the technical personnel (at least partially) of the positive cost-benefit trade-off of devoting time to user studies and usability [32]. We insist that attainable benefits should be clearly presented to the customer in order to win its buy-in of the principles of design thinking, its acknowledgement of the advantages of involving the users and its active collaboration in this. We point out to the research community that however, to this end, a set of actionable measures that can more objectively assess the positive impact of user involvement on the quality of produced software [18] is still lacking, together with a set of less resource-intensive practices to put such involvement in place.

Acknowledgments. Smart Campus was funded by TrentoRISE. The present work has been possible thanks to the funding granted by the Italian Ministry of Education, University and Research (MIUR) through the project “Città Educante”, project code CTN01_00034_393801. We wish to thank the Smart Campus team, the students who contributed to the project, and the Humus team for their kind support.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated.

The images or other third party material in this chapter are included in the work’s Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work’s Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

References

1. Ardito, C., Buono, P., Caivano, D., Costabile, M.F., Lanzilotti, R.: Investigating and promoting UX practice in industry: an experimental study. *Int. J. Hum. Comput. Stud.* **72**(6), 542–551 (2014)
2. Barrett, M., Oborn, E.: Boundary object use in cross-cultural software development teams. *Hum. Relat.* **63**(8), 1199–1221 (2010)
3. Beck, K., et al.: Manifesto for Agile software development. <http://www.Agilemanifesto.org>
4. Bjørn, P., Ngwenyama, O.: Virtual team collaboration: building shared meaning, resolving breakdowns and creating translucence. *Inf. Syst. J.* **19**(3), 227–253 (2009)
5. Bordin, S., De Angeli, A.: Communication breakdowns in the integration of user-centred design and Agile development. To appear. In: Cockton, G., Larusdottir, M.K., Gregory, P., Cajander, A. (eds.) *Integrating User Centred Design in Agile Development*. Springer, London (2016)
6. Bordin, S., Menéndez Blanco, M., De Angeli, A.: ViaggiaTrento: an application for collaborative sustainable mobility. *EAI Endorsed Trans. Ambient Syst.* **14**(4), (2014)
7. Brown, T.: Design thinking. *Harvard Bus. Rev.* **86**(6), 84 (2008)

8. Cajander, Å., Larusdóttir, 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)
9. Chamberlain, S., Sharp, H., Maiden, N.A.M.: 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)
10. Clark, H.H., Brennan, S.E.: Grounding in communication. *Perspect. Socially Shared Cogn.* **13**, 127–149 (1991)
11. Damodaran, L.: User involvement in the systems design process-a practical guide for users. *Behav. Inf. technology* **15**(6), 363–377 (1996)
12. De Angeli, A., Bordin, S., Menéndez Blanco, M.: Infrastructuring participatory development in information technology. In: *Proceedings of the 13th Participatory Design Conference: Research Papers*(1), pp. 11–20. ACM (2014)
13. Gothelf, J.: *Lean UX: Applying Lean principles to improve user experience*. O'Reilly Media Inc, Redwood Shores (2013)
14. Hussain, Z., Milchrahm, H., Shahzad, S., Slany, W., Tscheligi, M., Wolkerstorfer, P.: Integration of extreme programming and user-centered design: Lessons learned. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) *Agile Processes in Software Engineering and Extreme Programming*, pp. 143–153. Springer, Heidelberg (2006)
15. Jurca, G., Hellmann, T.D., Maurer, F.: Integrating Agile and user-centered design: a systematic mapping and review of evaluation and validation studies of Agile-UX. In: *Proceedings of Agile*, pp. 24–32. IEEE (2014)
16. Kujala, S.: User involvement: a review of the benefits and challenges. *Beh. Inf. Technol.* **22**(1), 1–16 (2003)
17. Lárusdóttir, M.K., Cajander, Å., Gulliksen, J.: The big picture of UX is missing in Scrum projects. In: *Proceedings of the 2nd International Workshop on The Interplay between User Experience Evaluation And Software Development, in Conjunction with the 7th Nordic Conference on Human-Computer Interaction* (2012). <http://ceur-ws.org/Vol-922/I-UxSED-2012-Proceedings.pdf#page=49>
18. Mao, J.Y., Vredenburg, K., Smith, P.W., Carey, T.: The state of user-centered design practice. *Commun. ACM* **48**(3), 105–109 (2005)
19. Memmel, T., Gundelsweiler, F., Reiterer, H.: Agile human-centered software engineering. In: *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as We Know It vol. 1*, British Computer Society, pp. 167–175 (2007)
20. Miller, L.: Case study of customer input for a successful product. In: *Proceedings of Agile*, pp. 225–234 (2005)
21. Miniukovich, A., De Angeli, A.: Computation of Interface Aesthetics. In: *Proceedings of the CHI*, pp. 1163–1172 (2015)
22. Nielsen, J. *Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier*. In: *Cost-justifying Usability*, pp. 245–272 (1994)
23. Nielsen, J.: *Usability Engineering*. Elsevier, New York (1994)
24. Nodder, C., Nielsen, J.: *Agile Usability: Best Practices for User Experience on Agile Development Projects*. Nielsen Norman Group, Fremont (2010)
25. Rogers, Y., Sharp, H., Preece, J.: *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, New York (2011)
26. Salah, D., Paige, R.F., Cairns, P.: 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*, p. 5. ACM (2014)

27. Schwartz, L.: Agile-User Experience Design: does the involvement of usability experts improve the software quality? *Int. J. Adv. Softw.* **7**(3&4), 456–468 (2014)
28. Sharp, H., Robinson, H.: Integrating user-centred design and software engineering: a role for extreme programming? (2004). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.99.4554&rep=rep1&type=pdf>
29. Smith, C.P.: *Motivation and Personality: Handbook of Thematic Content Analysis*. Cambridge University Press, New York (1992)
30. Sohaib, O., Khan, K.: Integrating usability engineering and agile software development: a literature review. In: *International Conference on Computer Design and Applications*, vol. 2, pp. V2–32. IEEE (2010)
31. Staiano, J., Menéndez, M., Battocchi, A., De Angeli, A., Sebe, N.: UX_Mate: from facial expressions to UX evaluation. In: *Proceedings of the DIS*, pp. 741–750. ACM (2012)
32. The Standish Group CHAOS report (2014). <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
33. Sy, D.: Adapting usability investigations for Agile user-centered design. *J. Usability Stud.* **2**(3), 112–132 (2007)
34. Teli, M., Bordin, S., Blanco, M.M., Orabona, G., De Angeli, A.: Public design of digital commons in urban places: a case study. *Int. J. Hum Comput Stud.* **81**, 17–30 (2015)
35. Ungar, J.M., White, J.A.: Agile user centered design: enter the design studio – a case study. In: *Proceedings of the CHI*, pp. 2167–2177. ACM Press (2008)
36. Venturi, G., Troost, J., Jokela, T.: People, organizations, and processes: an inquiry into the adoption of user-centered design in industry. *Int. J. Hum. Comput. Interact.* **21**(2), 219–238 (2006)
37. Wolkerstorfer, P. et al.: Probing an Agile usability process. In: *Proceedings of the CHI*, pp. 2151–2157. ACM Press (2008)