

Agile Human-Centred Design: A Conformance Checklist

Karsten Nebe^(✉) and Snigdha Baloni

Rhein-Waal University of Applied Science,
Friedrich-Heinrich-Allee 25, 47475 Kamp-Lintfort, Germany
Karsten.Nebe@hsrw.eu, snigdha.baloni@gmail.com

Abstract. Agile development is gaining popularity in industries in recent years. For more than a decade there have been researches about the integration of agile development and Human-Centred Design (HCD) so that organizations can benefit from the best of both. However, such approaches are being performed in-house and often not published broadly. What are best practices for the integration? Do these integration approaches affect the methodology of agile development? Or, are the principles of HCD being influenced by the aggregation of both methodologies? Agile human-centeredness needs to be assessed in relation to a reference model. Such a reference model did not exist. This paper provides a checklist that can be used to assess the user-centeredness of the agile processes. The checklist is based on DIN ISO 9241-210 in combination with best practices from agile human-centred design approaches and is evaluated by experts feedback.

Keywords: Agile development · User-centred design · Integration · Process assessment · Capability maturity model

1 Introduction

Software development frameworks have been evolving since there was a need to make software development more manageable and efficient [8]. Agile development processes were developed to overcome the recognized drawbacks of traditional sequential approaches [28]. Agile processes are iterative and incremental and lightweight. Another prevalent methodology is the User-Centred Design (UCD). UCD is a multi-disciplinary design approach, which emphasizes user involvement for clear understanding of requirements and tasks [14, 19]. UCD is based on iterations of design and evaluation. It is considered the solution to better user experience and usability. Various researches show that adoption of agile development is increasing day by day [2, 35]. In addition, some researches show that cost-benefit tradeoffs have become a key consideration for adopting UCD methods in product development [19]. Agile development processes allow delivering high quality products in a short period of time. On the other hand, UCD gives us the end user empathy to build more usable products with higher user satisfaction [26]. In order to achieve benefits from the advantages of both of these methodologies, a variety of researches have been done in the past few decades to integrate these two process models. Some of the significant integration strategies and frameworks are discussed in Sect. 2.2.

It is also observed that there is research data available for the assessment of agile processes and UCD processes, some of which are discussed in Sect. 3. Though there are various methods available to assess the agility of an agile process, there seems to be a lack of data available for assessing the user-centeredness of an agile process or vice versa, the agility of a UCD process. This paper explores possibilities to assess the ability of such organizations to perform human-centred design in agile development process environments.

2 Agile Development and HCD

Today, many industries have adopted agile development methodologies and the level of agile adoption is high and rising [2, 35]. The most popular agile approaches are Scrum and Extreme Programming [9, 20]. Compared to the traditional process models, agile methodologies are used for a dynamic environment where requirements, technologies and deadlines are changeable. Agile methods are structured keeping in mind that real world requirements are likely to change. Their basic values are interaction, collaboration and adaptability [17]. Agile models are considered as lightweight as they do not consider extensive documentation, and prefer informal and frequent communication over procedures [5, 17, 20, 27]. The iterative nature of agile development helps to concurrently work on smaller set of features to over the course of time form the bigger system. Large projects are therefore often implemented in series of small projects, delivering the customer a finished, tested and working implementation on a regular basis which can be more valuable to the customer than a document or a changeable overall design [1]. Agile development is not just about the agile values and practices but is also about managerial or organizational policies. Since all the values and principles are theoretical and there is no unique way to implement them, these can be perceived and implemented differently in different organizations [5, 27]. It's the core characteristics that are shared and the focus is on the basic principles of agile software development as explained in the Agile Manifesto.

According to Nielsen [23], the biggest risk of agile methods is that it is created by programmers or developers and focuses mainly on the implementation. Less importance or effort is given to usability. With the increase in the importance of user experience in system development for the past few decades, the need of more usable products has grown. User-centred design (UCD) is an approach that intends to make usable and useful products by keeping end-users in the center of the development process. The idea behind UCD is to gain a strong understanding of the user and the user's tasks and requirements [19]. This is done in an iterative manner with creating alternative solutions and testing them with users continuously. ISO 9241-210 [14] defines requirements and recommendations on human-centred approaches to design and shows how HCD can be effectively used and organized in a company's environment. As it is a management standard and it does not provide the methods and techniques needed for the implementation.

2.1 Comparison of Agile Development and UCD

Agile and UCD have the same goal of creating high quality products, but there are always differences in practices involved in each process. This is one of the reason integrating these two processes is a challenge for decades. The following aspects sum up the similarities along with the differences between the two perspectives:

- I. While agile and UCD are both iterative, there is difference in what is being iterated [3, 9, 27]. Agile methods iterate over application code or a smaller set of features that can be implemented. Iterations in agile are done during development. UCD iterates over user requirements and interaction design. Rannikko [27] states that a factor to keep in mind while integrating agile and UCD would be to make UCD incremental as well. In UCD practices, the user requirements are gathered and design solutions are being tested iteratively before the actual development begins. Making UCD incremental would mean to do user requirements elicitation and development of design alternatives for a short set of features iteratively. This makes it easier to be integrated with an agile process.
- II. Both the processes are user-centred or human centric. Agile and UCD principles all involve the user throughout design and development process. The difference here is that in UCD users are the end users of the product, i.e., users that would actually use the product, but agile processes rarely differentiate between customer and end-users [3, 27]. In order to integrate these two perspectives, there needs to be real end-user involvement in all the stages. In agile process there is no end-user involvement for evaluation of the design by definition. For agile to be more human-centred participation of end-users during designing and evaluation is needed to get feedback and achieve improvements continuously.
- III. Both the approaches emphasize effective teamwork and communication [3, 27]. Agile development process gives importance to face-to-face communications and keeping the team together, motivated and self-organized, which relates to the UCD principles as well. Another characteristic of UCD approach is that the whole team is supposed to design and develop keeping in mind the end-user [27].
- IV. Rannikko [27] points out that agile methodology works effectively in projects where requirements are not clear in the beginning. But agile methodologies do not mention how these requirements are being gathered. Agile teams depend on the customer for their requirements. In the case where the requirements are not clear to the customer, the team has to figure out by themselves and build something to ask the customer if the solution is what they expected. To save costs on changes of prior implementations, it is better to have done UCD practices before implementation. UCD methodology traditionally does up-front analysis, requirement gathering and design. Some high-level UCD practices, if integrated with agile, can help overcome such risks.
- V. In agile development developer based testing is done on a daily basis and apart from this testing there is customer/user acceptance testing that can be done after every iteration or Sprint (in Scrum), but it lacks any usability evaluation testing. Even though the developer and expert would be taking care of the usability, the real evaluation is complete only when it is tested with the end-users as well [27].

Evaluation in agile means acceptance from the customer who actually verifies if all the requirements are met and all functionalities implemented. There is less focus on usability of the system or product by default. UCD puts emphasis on the users and evaluates the system by testing with real users, which highlights usability defects and technical problems as well.

2.2 Agile and UCD Integration Approaches

Growths in research about agile and UCD integration have started to address the previously mentioned differences through different approaches. Past research showed how usability requirements were treated insufficiently in agile methods and which gaps need to be filled with UCD activities to make products more usable [5]. Silva et al. [32] did a systematic literature review of agile and UCD integration processes and propose an integration framework. This framework is almost similar to the approach that Fox et al. [10] came up with after interviewing 10 participants adapting or working in integration projects across America, Canada and Europe. Fox et al. came up with a Generalised, a Specialist and a Generalist-Specialist approach that was being used by most of the participants. Findings from their research show how the two methodologies are getting integrated in software industry. None of the processes followed by each participant was identical, but they had some commonalities. The authors constructed a high level representation of the similarities in the approaches and called it Agile-UCD General Process Model (AUGPM) [10]. Nebe and Paelke [21] also did a comparison of various kind of agile and UCD integration processes. They pointed out that every integration process has a distinct focus and does integration on one of the three levels, namely, on the level of Standards, Process Model or Operational process. The AUGPM developed by Fox et al. [10] is not only a very general approach, it also details the procedures to be followed. They not only confirmed the Sy model [33] but also showed that Sy's model is not limited to one organization. They have detailed the approaches by describing who performs which activities in the team. They also mention the output expected from every stage and every iteration and what should be the next steps, which was not provided in the details by Sy. Though AUGPM is a general approach that was designed with the similarities of processes followed by the participants of the study. Fox et al. also found distinct differences that they described as three different approaches for integration of UCD and agile. The approaches that emerge are different in the aspect that who executes which role in the process. These approaches are described with significant details. When looking into related work of integration of agile and UCD, it can be seen that most of approaches developed fit into one of the three approaches below proposed by Fox et al. [10]:

- The Specialist approach: In this approach, the team members are distinctly divided into UCDS (UCD Specialist) and development teams. There are three main groups, the users/customers, the UCDS, and the development team.
- The Generalist Approach: This approach uses two main groups, the user/customer and the developers that also act as UCDS. Developers that act as UCDS may or may not be formally trained UCDS. But they must have some informal or self-taught UCD skills. The roles that developers play are the main difference between the Specialist and Generalist approach.

- **The Generalist Specialist Approach:** In this approach there exists a team member that not only has formal UCD training but also has extensive software development experience. This member is a Generalist as well as a Specialist. Generalist because he can contribute in development task as well as perform UCD tasks and Specialist because he can also act as a UCD specialist.

3 Assessment of Agile- and UCD-Processes

A current state analysis is the first and typical step in a process to start any improvement endeavors in an organization. In software engineering and usability engineering process assessments are a widely used practice to this [16]. Datta [4] proposed a metric, the Agile Maturity Index (AMI) based on five software project dimensions, namely, Duration, Risk, Novelty, Effort and Interaction. The score obtained helps organization to decide the best fit methodology for a given project. Sidky [31] introduced the Sidky Agile Measurement Index (SAMI) which is based on the number of agile practices adopted by an organization. The “agile potential” of the organization depends on the number of agile practices being followed. SAMI has five agile levels and each level correlates with a set of principles and practices stated in the Agile Manifesto. Patel and Ramachandran [25] introduced in their paper the Agile Maturity Model (AMM) for agile software development environments. AMM is based on the agile software development values, practices and principles and is designed to improve and enhance the agile methodology being followed.

Apart from various research data and maturity models, there are many checklists and questionnaire available online that can be used for assessing the agility. For example, the Karlskrona test [30] provides an eleven-point questionnaire to analyze the agility of the existing process. The Agile Maturity Self Assessment consists of six questions based on the agile principles answered using the seven scale of Expertise [24]. Additionally there are the Nokia test [18] available online which is more specific to Scrum process, and the 42 point test: How Agile are You? [34] and the Scrum Master checklist [15]. These checklists mentioned are just a few from the many available online.

Various approaches have also been proposed and used to analyze and assess the current state of a UCD process. Earthy [6] introduced the Usability Maturity Model (UMM) to assess the level of maturity of an organization with regard to its capabilities to perform human-centred design. It is also known as the INUSE process assessment model [16]. This process assessment model is based on the software process assessment model defined in ISO 15504-2 in 1998 (This has been revised by ISO/IEC 33002 in 2015) [13]. The INUSE processes model consists of six levels of capability (Incomplete; Performed; Managed; Established; Predictable, Optimising). There are certain process attributes associated with each level of capability. These attributes determine whether a particular capability level is achieved. This process assessment model developed by INUSE project is well documented with all the details on how the assessment is to be performed. The process descriptions of the INUSE processes model was later published as a technical report ISO 18529 [12, 16]. Jokela [16] summarizes and contrasts various usability capability assessment approaches. He also came up with a new assessment

approach KESSU UPA. KESSU UPA consists of a new UCD process model, a three dimensional process-performance model and the implementation of the assessment as a workshop.

As shown, various researches and approaches to assess the agility of an agile process and also to assess the user-centeredness of a UCD process exist. Many frameworks and approaches have been already developed to get the benefits of both, some of which have been discussed already in this paper. But still, there is a lack of research data that would help in assessing the agile processes or the integrated agile-UCD processes for their user-centeredness [29]. Many organizations are willing and trying to integrate UCD in their agile development process these days. But therefore it is necessary to first assess the principle user-centeredness of the agile processes.

In the following section we explore the possibilities to measure the user-centeredness of an agile process applied in an organization, which would also help in knowing the capability of the organization using agile processes to perform HCD.

4 Checklist for User-Centredness of Agile Processes

The aim of this paper is to find a solution to measure the user-centeredness of a given agile process. It is relevant to mention here that this work is not aiming to assess an organization's process implementation maturity level. It attempts to assess the user-centeredness (as per the UCD values, principles and activities) of a given agile process model being followed (as per the agile principles, values and activities). For the purpose of this paper, it is important to have a basic definition for the process methodology in question that defines their principles, values and activities. The rest of the research is then performed depending on the basic definition of these process methodologies that are taken into consideration.

All developed processes are based on agile development methodologies principles, values and activities that are defined in the Agile Manifesto. The research work done for this paper also considers the Agile Manifesto for the basic agile definition while assessing a given agile process. To measure the user-centeredness of a UCD process there are various approaches available as discussed in previous section. The standard ISO/TR 18529 is one of them. As explained already, ISO/TR 18529 introduces the Usability Maturity Model, but it is yet to be investigated whether this model can be used for assessing user-centeredness of agile or integrated agile-UCD processes.

Here, ISO 9241-210 is chosen as reference model for assessing the user-centeredness of a process. With no assumption of any design process, it provides a HCD perspective that can be integrated into different design and development processes in a way that is applicable to the particular context. It describes how HCD can be planned and integrated into all phases of the product life cycle. The Annex B in the standards describes a sample procedure for assessing applicability and conformance. It provides a checklist that can be used to analyze whether HCD is being followed in a given process. Therefore, requirements and recommendations are listed in a checklist sequentially. The standard clearly states that the checklist is provided as guidance and must not be used as a substitute of the standard. Each requirement and recommendation is stated as a single point

in a separate row. Most of these requirements and recommendations have components. Every such component is stated as an entry in a separate row. Achievement of a requirement or recommendation depends on satisfying each of its components as well. Figure 1 displays a snapshot of this checklist showing the header attributes.

Clause/ subclause	Requirement or recommendation	Applicability		Conformance		
		Yes/No	Reason not applicable	Yes	No	Comments
4	Principles of human-centred design					
4.1	Whatever the design process and allocation of responsibilities and roles adopted, a human-centred approach should follow the principles listed [in 4.1].					

Fig. 1. Table B.1 (of ISO 9241-210) checklist header snapshot

To check the fulfillment of the requirements and recommendations, first the checklist is reviewed by experts to know which of these clauses are actually applicable for the project or process. This is documented by Yes/No in the third column with a justification in the fourth column, in case that particular clause is not applicable. Next, the clauses, which are marked as applicable, are then reviewed to determine whether the project or process conforms to those requirements and recommendations. How this is determined is not stated in the standard. It may vary from project to project and organization to organization.

For the intention of this paper the authors used this checklist as it defines HCD by principle and analyzed whether agile principles match or interfere.

4.1 Development of the Agile-HCD-Conformance Checklist

The checklist of ISO 9241-210 was analyzed for the purpose of this paper, exploring whether it would be possible to use this checklist to assess the user-centeredness of agile processes. To do so, each of the 103 entries in the checklist was examined whether it is applicable to agile development process (by definition as per the agile manifesto). This brought to light the gaps in agile development process, by definition, considering which the process can be made more human-centred. The analysis of the checklist was done in three steps: First, all 103 entries of the checklist were analyzed to find which entries or clauses of the checklist are actually valid enough when considering an agile development process. This was done to first eliminate the entries, which are completely unrelated to agile development. Second, the clauses in which a agile development process needed to be adapted in order to become human-centred were determined. Third, the gaps found in agile development processes were analyzed and recommendations as to how these gaps can be filled were provided. These recommendations are based on the state of the art research approaches (integration of agile and UCD as discussed in this paper).

During analysis of the checklist, it was found that all clauses of the checklist were valid enough to be used for assessing an agile process for its human-centeredness. All 103 entries of the checklist passed the first step of the analysis. In the second step of the analysis, 90 clauses could be found where agile processes need to be adapted in order to fulfill the requirement listed. By doing this, gaps in agile development processes (by definition) were determined. Lastly, the identified gaps were analyzed and a solution was given for every such clause. For example, consider the following clause from the

checklist: “Clause 4.3 (under point 4. Principles of Human-Centred Design): User involvement should be active.” This particular checklist clause is valid to be used for the assessment of an agile process. But when the HCD principles are considered, a gap is observed in agile processes even though agile methodology does involve users/customers (see 2.1.II). In HCD, user involvement should be active throughout the project life cycle and especially during the early stages. As defined in ISO 9241-210, the real end-users must be involved during requirements gathering where, if needed, low fidelity prototypes must be included to get more definite user requirements and understanding of the users tasks. Moreover, in HCD end-users must also be involved during designing. The designs must then be tested with real users to get their feedback for improvements before the actual implementation starts. In the end, a user-centred evaluation must be performed as the final step after implementation is done. As per the agile definition, the first point to note is that agile does not differentiate between users and customers [3, 27]. Second point to note is that agile processes does not emphasize on user involvement during the early stages and user evaluation during the design phase. Following this procedure the analysis highlighted gaps in agile development processes. Then, recommendations for improvement could be given. For the above example, it is recommended to involve real users in the early stages (e.g. Sprint Zero) as well as during the agile development process, especially by the involvement of real end users during requirements analysis and elicitation. In addition, the user needs to also be involved to evaluate the designs apart from just testing the implementation at the end [7, 10].

Every entry in the checklist was analyzed as per the process explained above. This resulted in the development of a new extended version of this checklist having the same clauses as in ISO 9241-210 checklist but with agile gap findings and additional recommendations for implementation of HCD. This extended list can help filling the gaps and makes agile processes human-centred. This Agile-HCD-Conformance Checklist can be found at the URL [22]. The extended checklist is quite similar to the ISO 9241-210 checklist with a slight difference. Columns one and two remain the same, i.e., first being the clause or sub clause number and the second the requirement or recommendation. The third and fourth column contains information related to the applicability of that clause to agile process (by definition). The third column is filled with Yes/No depending whether that recommendation or requirement is applicable to the agile process according to the agile definition. In case a recommendation or requirement is not applicable, the reason is mentioned in the fourth column. These basically determine the gaps. The fifth and the last column contains the recommendations to make the agile process more human-centred based on the related work already done in integration of agile and UCD.

4.2 Evaluation of the Agile-HCD-Conformance Checklist

To evaluate if the checklist could be used to assess an organization’s ability to perform human-centred design in an agile development processes and whether it provides means for the implementation, expert feedback was needed. Experts must have a strong knowledge of agile development processes, based on their educational background or work experience of years in agile development projects along with having certifications related to agile processes (like the Scrum Master certification).

The statements, which are present in the “Applicable to agile process (by definition)? (Y/N)” and “Reason not applicable” columns of the checklist, are the entries of the checklist, which were needed for the evaluation through experts. These describe the gaps of human-centeredness in agile processes, which need to be filled. Therefore an online evaluation questionnaire was created (*For future studies input is still needed. Please help and support by providing expert input at [22]*). Each question of the questionnaire consists of the specific requirement or recommendation as in the original checklist, followed by the agile applicability statement from the developed new checklist. Every question was comprised of two parts. One was the requirement or recommendation clause and the second was the “Applicable to agile process (by definition)? (Yes/No)” and the “Reason not applicable” statement. The participants were asked to give their feedback for each agile applicability statement, i.e., whether they agree or disagree with the given statement (multiple choice). They were also provided with separate text boxes to write their comments after every question.

The online questionnaire was sent to 12 experts. Five responses were received. By looking at the results, similar trends could be identified. The evaluation report with the responses of the participants along with their comments can be found at the URL (Nebe [22]). In summary it shows that the experts mostly agree with our findings.

Keep in mind, while analyzing and proposing the Agile-HCD-Conformance Checklist, the purpose of this research was not to analyze for operational process implementation, but was a focus on fundamental agile development principles. Some of the experts’ responses unconsciously compared the checklist entries with the given agile process they were following in their organization. Such responses were excluded. Also, not all negative responses from experts meant that they disagree with our checklist statement. In few situations, experts state a negative response but after reading their comments it was understood that they actually disagreed with the actual checklist statement for agile, which was what expected and explained in our “reasons not applicable” section. In some few cases it was seen that our “Reason not applicable” statement is not clear enough and the responses from the experts helped us to refine “Reason not applicable” statement in the new version of our checklist.

Altogether it can be said, that the Agile-HCD-Conformance Checklist can be used as a good source for the assessment of the user-centeredness of agile processes and can be used as a guideline for setting up new agile HCD processes.

5 Summary, Conclusion, Discussion and Outlook

This research explored approaches of assessment for agile processes and human-centred design (HCD) processes. It was important for this work to understand the underlying basic principles and values of both the methodologies. In addition, state of the art research about strategies for the integration of agile and UCD are reported. Even though, many integration approaches exist, there is still a gap of principal recommendations and practices for the integration of agile and HCD processes. The authors present an approach that give guidance for such integration based on principles and best practices

of agile processes and user-centeredness. The standard ISO 9241-210 was chosen to explore the possibilities of assessing the user-centeredness of agile processes. Based on this, the Agile-HCD-Conformance Checklist [22] was developed so that it can be used as a source for the assessment of the user-centeredness of agile processes. The Agile-HCD-Conformance Checklist was evaluated with experts. Minor refinements were done after the evaluation results analysis. It is concluded that the research was successful in finding the gaps that would help the agile process to adapt and perform HCD. The Agile-HCD-Conformance Checklist, also provides recommendations on how the gaps found in agile process can be filled. Recommendations are given based on state of the art researches in the field of integration of agile and UCD.

Due to the limitation of time, the evaluation is based on feedback of five experts. Given the time, more feedback could have been gathered that might have helped to refine the findings more and make it more concrete. *If you are interested in supporting this approach and future work, please provide your expertise in filling the questionnaire [22]. Thanks a lot.*

This research leads to a vast area for research questions in the area of measuring the user-centeredness of an agile process or an integrated user-centred agile process. Moreover, it also leads to questions in the area of measuring the agility of a user-centred design process or an integrated agile user-centred process. Further research can also be done to develop a maturity index to analyze whether an agile process can be put in a level of maturity in terms of user-centeredness or vice-versa.

Acknowledgements. We are very grateful to all the participants in this study without whom this analysis would not have been possible, and to Jonathan Yen for his support in finalizing this paper.

References

1. Armitage, J.: Are agile methods good for design? *Interactions* **11**(1), 14–23 (2004)
2. Bustard, D., Wilkie, G., Greer, D.: The maturation of agile software development principles and practice: observations on successive industrial studies in 2010 and 2012. In: 20th IEEE International Conference and Workshops, pp. 139–146. IEEE, Scottsdale (2013)
3. Chamberlain, S., Sharp, H., Maiden, N.: Towards a Framework for Integrating Agile Development and User-Centred Design. <http://www.ime.usp.br/~marivb/ihc3.pdf>
4. Datta, S.: Agility measurement index – a metric for the crossroads of software development methodologies. In: Proceedings of the 44th Annual Southeast Regional Conference, pp. 271–273, New York (2006)
5. Düchting, M., Zimmermann, D., Nebe, K.: Incorporating user centered requirement engineering into agile software development. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4550, pp. 58–67. Springer, Heidelberg (2007)
6. Earthy, J.: Usability Maturity Model: Human Centredness Scale. <http://www.idemployee.id.tue.nl/g.w.m.rauterberg/lecturenotes/usability-maturity-model%5B1%5D.pdf>
7. Ferreira, J., Nobel, J., Biddle, R.: Agile development iterations and UI design. In: Proceedings of Agile 2007, pp. 50–58. IEEE, Washington (2007)

8. Fowler, M.: The New Methodology. <http://www.martinfowler.com/articles/newMethodology.html>
9. Fox, D.: Agile Methods and User-Centred Design: How These Two Methodologies are Being Integrated in Industry (M.Sc. thesis), University of Calgary (2010). http://pages.cpsc.ucalgary.ca/~maurer/uploads/Publications/Agile_UCD.pdf
10. Fox, D., Sillito, J., Maurer, F.: Agile methods and user-centred design: how these two methodologies are being successfully integrated in industry. In: AGILE 2008, Conference, pp. 63–72. IEEE, Toronto (2008)
11. Iver, R.M., Campbell, B.: Accelerating your organization's agile adoption. In: Agile 2010 Conference, Orlando (2010). <http://www.robbiemaciver.com/documents/presentations/A2010-Agile%20Maturity%20-%20Presentation.pdf>
12. ISO/IEC, ISO/TR 18529 Ergonomics – Ergonomics of Human-System Interaction – Human-Centred Lifecycle Process Descriptions (2000)
13. ISO/IEC, ISO/IEC 15504-2 Information Technology – Process Assessment – Part 2: Performing an Assessment (2003)
14. ISO/IEC, ISO 9241-210 Ergonomics of Human System Interaction-Part 210: Human-Centred Design for Interactive Systems (2010)
15. James, M.: A ScrumMaster's Checklist. <http://www.scrummasterchecklist.org/>
16. Jokela, T.: Assessment of User-Centred Design Process As a Basis of Improvement Action. <http://herkules.oulu.fi/isbn9514265513/isbn9514265513.pdf>
17. Lan, C., Ramesh, B.: Agile software development: ad hoc practices or sound principles? IT Prof. **9**, 41–47 (2007). IEEE
18. Little, J.: The Nokia Test. <http://agileconsortium.blogspot.com/2007/12/nokia-test.html>
19. Mao, J., Vredenburg, K., Smith, P.W., Carey, T.: User-centred design methods in practice: a survey of the state of the art. In: Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research, p. 12. IBM Press (2001)
20. McInerney, P., Maurer, F.: UCD in agile projects: dream team or odd couple? Interactions **12**, 19–23 (2005)
21. Nebe, K., Paelke, V.: Usability-engineering-requirements as a basis for the integration with software engineering. In: Jacko, J.A. (ed.) HCI International 2009, Part I. LNCS, vol. 5610, pp. 652–659. Springer, Heidelberg (2009)
22. Nebe, K. (2016). <https://www.hochschule-rhein-waal.de/de/fakultaeten/kommunikation-und-umwelt/organisation/professoren/prof-dr-karsten-nebe/publikationen-0>
23. Nielsen, J.: Agile Development Projects and Usability, Nielsen Norman Group, on 17 November 2008. <http://www.nngroup.com/articles/agile-development-and-usability/>
24. Page-Jones, M.: The Seven Stages of Expertise in Software Engineering. <http://www.wayland-informatics.com/The%20Seven%20Stages%20of%20Expertise%20in%20Software.htm>
25. Patel, C., Ramachandran, M.: Agile maturity model (AMM): a software process improvement framework for agile software development practices. OALib (2014). <http://www.oalib.com/paper/2761472#.VAmwFcKSYSo>
26. Patton, J.: Hitting the target: adding interaction design to agile software development. In: Proceedings of OOPSLA 2002 OOPSLA 2002 Practitioners Reports, pp. 1-ff. ACM, New York (2002)
27. Rannikko, P.: User-centred design in agile development. M.Sc. thesis, University of Tampere. <http://tampub.uta.fi/bitstream/handle/10024/82310/gradu04854.pdf?sequence=1>
28. Salah, D.: A framework for the integration of user centred design and agile software development processes. In: 2011 33rd International Conference on Software Engineering (ICSE), pp. 1132–1133. IEEE, Honolulu (2011)

29. Salah, D., Paige, R., Cairns, P.: Integrating agile development processes and user centred design- a place for usability maturity models? In: Sauer, S., Bogdan, C., Forbrig, P., Bernhaupt, R., Winckler, M. (eds.) HCSE 2014. LNCS, vol. 8742, pp. 108–125. Springer, Heidelberg (2014)
30. Seuffert, M.: Karlskrona test. <http://mayberg.se/learning/karlskrona-test-online>
31. Sidky, A.: A structured approach to adopting agile practices: the agile adoption framework. Ph.D. dissertation, Computer Science, Virginia Tech, Blacksburg (2007). <http://scholar.lib.vt.edu/theses/available/etd-05252007-110748/>
32. Silva, D.T., Martin, A., Maurer, F., Silveira, M.: User-centred design and agile methods: a systematic review. In: Agile Conference (AGILE) pp. 77–86. IEEE, Salt Lake City (2011)
33. Sy, D.: Adapting usability investigations for agile user-centred design. *J. Usability Stud.* **2**, 112–130 (2007). <http://uxpajournal.org/wp-content/uploads/pdf/agile-ucd.pdf>
34. Waters, K.: How Agile Are You? (Take This 42 Point Test). <http://www.allaboutagile.com/how-agile-are-you-take-this-42-point-test/>
35. Williams, L.: What agile teams think of agile principles? *Commun. ACM* **55**(2), 71–76 (2012)