Integrating Usability Engineering and Agile Software Development:

A Literature Review

Osama Sohaib
College of Computing and Information Sciences
PAF-KIET
Karachi, Pakistan
osama.sohaib@gmail.com

Khalid Khan
College of Computing and Information Sciences
PAF-KIET
Karachi, Pakistan
khalid.khan@pafkiet.edu.pk

***Abstract* — The various agile software development methodologies have promoted since their inception and even demanded high degree to improve the quality of the software product. Usability engineering has made its way into the software mainstream and has caught the attention of software engineers and researchers worldwide due to rapidly growing and volatile internet software industry, despite their different perspectives on creating software both have a major role in making good software. Usability focuses on how the end users will work with the software and agile development focuses on how the software should be developed. The way these two perspectives are being combined in practice is still not well understood. This study is a preliminary literature review that describes the key question that how usability-engineering practices should be integrated with agile software development in order to make stronger and effective usable software system. This paper focuses on identifying the tensions between usability and agile methods. The research aim is to identify the common approach of agile methods and usability engineering by surveying extensive related work on integration of usability and agile methods.**

Keywords – usability; user-centered design; agile development; extreme programming (XP);

I. INTRODUCTION

Software plays a significant role in lives of persons and in severe competition between companies. That is why Software Engineering (SE) is a discipline, which is not shy of introducing new methodologies, and is clearly maturing on complex projects with great achievements. In the past few decades, the issue of how software process development and improvement has been discussed that how it should organize in order to deliver faster, better, and cheaper solutions. Large numbers of different approaches to software development have introduced, of which only few have survived for used today. In recent years, so-called agile methodologies have become increasingly popular. These processes are intended to support iteration rather than traditional development methodology phases and focusing on continual interaction between developers and customers [1]. After the completion of iteration, the project team discusses and seeks solutions of the problems discovered in the last iteration. Many of the suggestions for improvement have come from experienced practitioners, who have labeled their methods agile software development. While the elements of usability engineering model according to Nielson [2] is early focus on users, user participation in the design, coordination of the different parts of the user interface, empirical user testing, and iterative revision of designs based on the test results. Although agile methods share, some of the same aims as usability but there are also distinct differences. From the perspective of usability and user-centered design, however, agile methods do not inherently provide the required support to the development process. In fact, the agile and user-centered approaches have the potential to work very well together. Many researchers have discussed the integration of usability and agile methods. This paper tries to combine these finding into a cogent wholeness by addressing the following research questions.

RQ1. What tensions between usability engineering and agile methods have been identified in related research that makes them difficult to integrate?

RQ2. What approaches have been suggested in order to integrate usability and agile methods?

The work is organized as follows. Section 2 gives an overview and background of the agile development and the need for usability. Section 3 presents extensive related studies about how usability and agile might work together. Section 4 provides some insights from the literature review and presents an integrated approach. Finally, the study is concluded and leaves an open issue.

II. BACKGROUND

In this section, the literature is used to present some background on agile development and on the need for usability.

A. Agile development overview

The term agile can be defined as "characterized by quickness, lightness, and ease of movement; nimble" (dictionary.com) the term lightweight is also used for agile and has increased the public attention in late 1990's [3]. One of the primary causes of software project failure is the extended period of time it took to develop a system. Agile methods were developed to develop systems more quickly with limited time spent on analysis and design. Agile

methods are iterative, focus on teamwork, collaboration between customer and developer, feedback from customer throughout the lifecycle of the software project and support early product delivery [4]. Intention is to make the development simpler and serve the customer better. Traditional software development approaches is not always feasible in today rapidly changing environment however agile methods promote possible quick response to changing environment. Waterfall model one of the traditional software process models where any stage should not start until the last stage is complete while agile methods based on various iterations to analyze, design, implement and test to release the product [5]. Agile methods do have practices that have quality assurance abilities, some of them are inside the development phase and some others can separated out as supporting practices [7].

To help people to gain a better understanding of what agile software development is all about, the members of the Agile Alliance refined the philosophies captured in their manifesto into a collection of twelve principles. The twelve principles behind agile manifesto are [3].

- The highest priority is to satisfy the customer through the early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from every few weeks to every few months, with a preference for the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build the projects around motivated individuals. Give them the environment and support they need, then trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done is essential.
- The best architectures, requirements and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, and then readjusts its behavior accordingly.

McBreen [6] defines the agile software quality as the development of software that can respond to change, as the customer requires it to change. This implies that outcome at the end of each phase is sent frequently the delivery of tested, working, and customer-approved software to quality assurance practitioners for feedback. In order to speed up the product delivery processes the agile development practice and software quality methods cooperating with each other throughout the project life cycle.

Huo [7] describes *System metaphor* is used instead of formal architecture, which presents a story that how a system works by describing classes and pattern of the system, which is helpful for communication between customer and developer and help the agile development team in architecture evaluation. While the *onsite customer* is the common practice in most agile projects which helps the developer in refining requirements throughout the project by communication. In traditional software development approach, customer is only involved during requirement gathering but much is expected in agile methodology. *Pair programming* helps in reducing defects in code where two or more programmers working on the same code. *Refactoring* is the essential part of the agile methods that is the process of changing internal structure without changing the external behavior to help in understandability of the code. During refactoring developers reconstruct the code and this action provides code inspection functionality. This activity reduces the probability of generating errors during development. *Continuous integration* occurs several times a day, it saves the time of the developer spend on searching for bugs as the system is kept integrated all the time, several times a day. *Acceptance testing* occur early rather than other traditional process models and one of the most uniqueness of the agile methods is early customer feedback which provides information to the development team early in the project lifecycle.

B. *The need for usability*

"Usability Engineering" is a science that studies how to understand and systematically address the usability demand of a customer [8]. The ISO 9241-11 defines usability as *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"*. Usability engineering deals with issues such as system learnability, efficiency, memorability, errors and user satisfaction [2]. Its processes focus on developing systems that are adapted for end users. Its underlying practices and theories can give clear idea about user motivations, characteristics, work environments, and depict on many different areas including psychology, sociology, physiology and human factors [8]. The ISO 13 407 Usability Model is comprised of five stages; the model is graphically depicted in Fig. 1 [9]. The Four basic activities in the design process are

1. Identify needs and establish requirements
2. Design potential solutions (re-design)
3. Choose between alternatives (evaluate)
4. Build the artefact

Usability engineering is an approach to product development that is base on customer data and feedback on direct observation and interactions with customers to provide more reliable data than self-reporting techniques. Usability engineering begins in the conceptual phase with field studies and contextual inquiries to understand the

functionality and design requirements of the product [10]. Usability engineering provides structured methods for achieving usability in the system development process. It is an iterative design and evaluation to provide customer feedback on the usefulness and usability of a product's functionality and design throughout the development cycle. This results in products that are developed to meet the customers' needs [10].

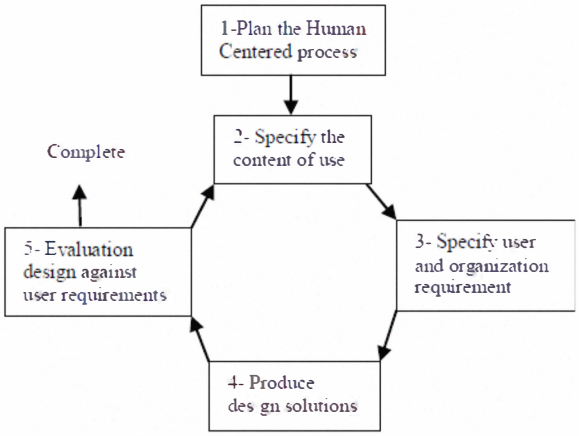

Figure 1. ISO 13 407 Model Overview [8]

A key methodology for carrying out usability is called User-Centered Design (UCD) [11]. Quality of a user's experience (UX) is measured by usability when interacting with system whether a web site, a software application, or any user-operated mobile device [11]. According to Vredenburg et al. [12], UCD is *"An approach to designing ease of use into the total user experience with products and systems. It involves two fundamental elements – multidisciplinary teamwork and a set of specialized methods of acquiring user input and converting it into design"*. The following are the six core UCD principles, which correspond to the heart of UCD and provide as the structure for individual methods and procedures [12]. Set business goals, Understand users, Design the total customer experience, Evaluate designs, Assess competitiveness and Manage for users. Typical UCD activities by Usability Professional Association (UPA) are broken down into four phases: Analysis, Design, Implementation and Deployment, with suggested activities for each phase [13].

III. RELATED WORK

Some recent research examines usability and agile methods to determine, if and how these two approaches integrated in the development process. In recent years, agile practitioners have begun to explore ways of incorporating usability into agile methods. Based on an analysis of core values and principles of both areas of agile development and usability practices, an approaches proposed by the agile community that contributes to the tensions between usability and agile methods. However, there was some agreement on integration of both usability and agile methods but still there are some issues and challenges.

A. Customer Focus Vs. End-User

In agile software development, the customer is a role with several responsibilities, amongst others the provision of requirements. The customer should sit with the team, write and priorities user stories and test the software. Due to this understanding of the customer, participatory design and collaboration do not equal user involvement, so there is a question of whether they represent typical users. In order to design something to support people, we must know who our target users are and what rather support an interactive product could usefully provide.

Many software development organizations are reporting great success with agile software development techniques, however, none of the major agile development methods explicitly incorporates usability-engineering practices, and e.g. they may not address the needs of novices as well as expert users. Agile methods alone do not ensure that application UI is usable. As more organizations adopt usability and Agile development practices for software products, usability practitioners are not sure of, that the resulting products are designed with users in mind. Agile methods share some of the same aims as user centered design (UCD), but there are some issues that make them difficult to integrate, such as communication issues, time differences between designers' and developers' and these do not equal user involvement. Agile models have significant deficiencies in handling user-centered requirements where usability requirements are treated insufficiently in the important stages of development. There are other challenges in including usability techniques in agile development, as it has multiple iterations and often substitutes the client as the user-advocate. Software engineers that practice Extreme Programming (XP) value the "input from the on-site customer", are not using the concept of user-centered design [14], [15], [16], [17], [18], [19], [20], [21], [22], [23].

Kane [14] discusses that none of the major agile development methods explicitly incorporates usability-engineering practices, and suggests that by incorporating discount usability engineering techniques as part of an agile development methodology can improve the usability of software. There are still many issues that would have to be addressed to make the combination effective.

Fox et al [15] addresses that agile methods alone do not ensure that application user interface (UI) is usable, and suggested an approaches to achieve the integration are the Generalist, Specialist, and the Hybrid approach. All of the approaches did have at least one UCD's or a team member acting as a UCD's. In order for agile and UCD methodologies to be integrated both must compromise their upfront resource allocation.

Sy [16] also describes the process of integrating UCD with agile methods. Sy describes the main adaptations to the timing, granularity, and reporting used for agile interactive usability investigations, with an intended audience of usability practitioners. The author found that the new agile

UCD methods produce better-designed products than the "waterfall" versions of the same techniques. What does she left a future challenge is that how to capture the closeness and clarity of the storytelling with team members in other locations.

Unger and White [17] describe the merger of user-centered design into agile team development practice as a manifest in a one-day design studio. The design studio approach meant the entire team members gain enough of a shared understanding of the design to begin development and strengthening the collaboration between designers and developers.

Najafi and Toyoshiba [18] describes incorporating User Experience Design (UED) practices in agile development to improve product usability by involving the User Experience team, by doing so, user research and testing can be utilize to prioritize features in the product backlog and to iteratively refine designs to achieve better usability. They recommend that successful integration of the User Experience team require full cooperation and collaboration with all cross-functional team members.

Chamberlain et al [19] have conducted a field study to explore a framework for understanding how interaction design and agile development might work together. Their study identifies challenges and the broad issues on which the two processes are similar, including the need for communication between and participation of both practitioners on a project. The evidence from their observations suggests that it will be more likely to deliver benefits to the business and most importantly to the user.

Eklund and Levingston [20] offers a set of recommendations for those IT organizations that concerned about how to include usability testing in a agile software development, such as to involve users at several stages of development and the engagement of an external usability specialist to give advice as required. However, the recommendations have to be test in real projects.

Düchting et al [21] analyzed Scrum and XP process to see how they are able to adopt UCD activities and introduced a requirement engineering framework to determined whether the two agile SE Models (Scrum and XP) consider the three types of requirements (Usability requirement, workflow requirement and the User Interface Requirements). Using and applying them to other agile models will enable to derive more generic statements about the integration of usability engineering (UE) in agile SE models in general.

Memmel et al [22] present a CRoss-discipline User Interface design and Software Engineering (CRUISER) lifecycle that integrates SE and HCI under the umbrella of agile development. CRUISER is based on XP; the result shows using scenarios and prototypes as fundamental artifacts drive a design process with high involvement of users and stakeholders.

Sharp et al [23] introduced customer collaboration in XP, UCD principles are compared and their similarities indicate that UCD techniques may be useful and welcomed. For example, using participatory design, or including heuristic evaluation in each of iteration. The main issues the authors discovered are; trust between developers and customer, Software testers may make good bridges, modeling was suggested as form of bridge, Access to the right people and face- to face collaboration of project stakeholders via workshops. However, customer may not be the real user [23].

B. Working Software Vs. Usable Software

Agile development aims to start delivering working code as soon as possible. However, particularly in XP the overall structure of the product (including user interface) is not design before the implementation is started. Instead, XP focuses on implementing only a little piece of functionality in each of iteration. Continuous integration, refactoring and unit testing guarantee that this does not lead to inconsistent code, but it is not easy to guarantee that the user interface does not become inconsistent. One of the main principles of XP is that working software is the primary measure of progress. This is a common goal with XP and UCD since in order for the software to be usable it has to be working. However, working software does not equal usable software. In XP, working software emphasizes the quality of the code, which may in fact decrease the developers' motivation to focus on the usability issues. With an increasing number of software development projects that follow an agile approach, User Experience (UX) practitioners are not well understood that how user-centered design (UCD) techniques can be applied in an agile context. The Extreme Programming delivers high quality software efficiently but the resulting software might fail to delight the customer. Usability engineering methods are not integrated into agile methods when developing UI intensive systems [9], [24], [25], [26].

Lee and McCrickard [9] offers an agile usability development process and toolset through applying extreme programming (an agile software development process), and scenario-based design (a usability engineering process) to allow developers in both fields to better communicate and work together to efficiently design usable systems. According to authors, there is still a need to develop ways to design software systems that can draw on the best practices and tools of different disciplines.

Kollmann et al [24] proposed a qualitative approach that identifies two main themes (identity and vision) that perceived by UX practitioners to be highly important in the success of integrating UCD and agile approaches. These two themes are; UX practitioners' understanding of their job role, and the need to establish, protect and communicate an overall team vision. Still some issues related to co location of designer and developer needs to be address.

Patton [25] discusses the adapting UCD practices into his agile development process (eXtreme Programming (XP). He describes the ten-step process that has added to their current development process to produce an agile usage centered design process. The author does not describe how the design was passed from the design stage to the implementation stage.

Lee et al [26] also proposed an integrated approach known as eXtreme Scenario-based Design (XSBD). Four requirements are identified that need to be met for an integrated approach to work effectively. The key design representation in the XSBD process is the central design record (CDR), which is used to support synchronization activities, help the usability engineer plan, and run usability evaluations.

C. Required Design Vs. Up front Design

UCD processes rely on up-front user design. Agile methods, on the other hand, aims to start design whenever required by the customer. While agile practitioners do model, they try to avoid an extensive up-front design phase. Many agile projects require user interaction (UI) design, but the integration of UI design into agile development is not well understood [27].

Ferreira et al [27] investigate four different projects and identifies that agile iterations facilitates usability testing. He also argues about the Big Design Up Front" (BDUF), suggesting that the more the design is determined up front, the more difficult it is to change later on, so does the agile development care about BUDF apply to UI design? However, they address this issue in [28] and it was found that up-front interaction design is a common place in agile development there are benefits to interaction design being done in the iterations together with the software development, most importantly, it reduces risks. However, the integration issue still needs to be addressed.

D. Unit Testing Vs. Usability Testing

Although testing is emphasized in agile methods, In XP, especially unit testing and acceptance testing where each individual module is tested but it contains no practices that directly support usability testing. There is no time to do thorough usability tests with users between iterations or release cycles and only testing paper prototypes and doing expert analyses do not provide an accurate picture of the product's usability.

Multi-disciplinary system designs are found deficient due to lack of usability evaluation in agile development. Agile development and user centered design shares many common principles but it is complicated for agile methods to incorporate user centered design practices such as low fidelity prototyping. In order to increase software quality Usability Engineering and Software Engineering often work together However, frictions between organizational and technological constraints can lead to difficulties when the interaction of both methodologies is limited. Ad-hoc input, Cultural and onsite customer representatives are the major issues to prevent integration of HCI instrument into to the Extreme Programming (XP) processes. Usability engineering and extreme programming employ similar methods to achieve their goals but there is lack of production model in XP, such as paper mock-up and prototype and the user stories may not be good enough from the usability perspective in SCRUM [29], [30], [31], [32], [33], [34].

Lee [29] describes that how effectively the Central Design Record (CDR) allowed developers to address interaction architecture issues in an agile development environment. The proposed solution is that by extending the concept of test-driven development to user interface testing and integrating it with the CDR design representation, developers will be able to validate system usability and iteratively enhance designs without requiring extensive up-front requirements analysis. The author describes the difference between the usability test suite and the code test suite is that the usability test suite cannot entirely automate since many evaluation techniques require participant data and feedback. Still Many existing agile development processes are unable to account for continuous requirements and system changes requested throughout the development process.

McNeill [30] argued that user centered design fits well with agile methods. He describes that by incorporating user-centered design in particular using low fidelity prototyping and storyboarding through collaborative workshops as an iterative model for the application rather than time-consuming code, better applications are develop and will deliver business benefit with a focus upon the end user and their experience.

Obendorf and Finck [31] proposed an approach using the scenario techniques in Extreme Programming that preserve the agility of agile development methods. He addresses and argues for scenarios as tools for combining XP and Usability Engineering and reflects on the experiences they have made adapting it to real projects. However, the integrated approach needs more reflection.

Hussain et al [32] presented the context where the HCI instruments (user studies, extreme personas, usability expert evaluations, usability tests, and automated usability evaluations) are utilized in a mobile multimedia application. They have combined the advantages of the Extreme Programming methodology (on-time delivering, optimized resource investments, short release cycles, working high quality software, tight customer integration) with the advantages of a user centered design process (usable, accessible, and accepted products, end-user integration).

Holzinger et al [33] present how to combine Extreme Programming (XP) and Usability Engineering (UE) in an even more powerful development method called Extreme Usability (XU). Extreme Usability (XU) could become such that all the best practices of UE are keep in the XP process during the planning games including think aloud method on paper mockup.

Singh [34] proposed a U-SCRUM methodology specifically to help improve usability. Unlike typical SCRUM, U-SCRUM is having two product owners, one focused on usability and the other on the more conventional functions. Moreover, the result shows that U-SCRUM yields improved usability than SCRUM. However, further studies are needed if it has multiple usability specialists for complex products.

This work shows the continuing efforts by exploring the combine practices between agile software development and usability. There is a still need to develop ways to design software systems that can draw on the best practices and tools of different disciplines. Agile software development and usability engineering both addresses important aspects of software development and due to the iterative nature of agile methods, UCD is well compatible.

McInerney and Maurer [35] have already showed that the integration of two approaches is possible. Software engineers and usability experts come from different domains with different attitudes and most importantly agile development focus is on technical functionality ignoring usability issues.

Many leading software companies are adopting usability in agile development, but there is still a difficulty and lack of tactics and practices in agile development to develop and maintain sound user centered design (UCD) practices within organization when moving to agile development.

In this study, related literature and research of both detailed implementation of usability engineering and agile methods approaches are used to determine a generalized and a combined approach that contribute to the tensions between usability and agile methods.

IV. RESULTS

For a better understanding of how usability and agile development might work together, some observations from the literature about their similarities and differences are collected for a combine approach.

The major challenge for an agile approach is how to identify the requirements of a system as accurately as possible from a customer who is not the actual end user. Usability engineering requires various upfront modeling and scenario based design alternatives before a best design is selected for implementation while agilest believe that these artifacts do not deliver any value to the customer and follows user stories for capturing requirements but does not capture all the details. The participatory and iterative aspects of usability engineering are similar to the agile approach both follow evolutionary system development. Both uses test driven development, in agile development testing is perform throughout the project lifecycle while usability engineering use various testing techniques for example Think aloud, Performance measurement, wizard of Oz and field surveys etc. Each of them used prototyping. In usability engineering, design begins with throwaway prototyping while agile follow throw away and evolutionary prototyping. Agile and usability both support strong collaboration between different stakeholders involved in the project. Face to face, communication and working in pair are preferred. Another point is the generalist/specialist aspect; agile teams are ideally generalists [14], i.e., they possess a wide range of skills rather than one narrowly defined area of expertise while usability requires usability specialist. A combined approach of usability and agile methods based on the observation from the literature is summarized in Table 1.

Table 1: Combine approach of usability engineering and agile methods

Agile Methods Concepts	Usability Engineering	Suggested Approach
Deliver working software frequently	Traditional software approach but iteration within phases	Iterative development throughout the project
Requires generalists	Requires specialists	Assemble a multidisciplinary team to ensure complete expertise
Customer focus	User focus	Collaboration between customers, users, product managers, Business analysts, developers, will maximize overall team efficiency for usable product.
Test driven development and continuous integration	Contextual inquiry, field surveys, usability inspection methods for testing	Unit Testing + User Acceptance Testing + Usability testing throughout the process
Using onsite customer, functional requirement are encapsulated as user stories	Scenario based design for requirement analysis	Integrate user stories with scenario based design

V. CONCLUSION AND FUTURE WORK

In Conclusion, this paper has argued that usability fits well with the agile software development. Usability engineering can be adapted to the agile context, for example by using a more iterative approach and testing throughout the project lifecycle. Agile projects needs to adopt aspects of usability engineering by incorporating user scenarios and including usability specialists in the team. Agile development arguing for quickly moving towards the source code while the usability suggests waiting with implementation activities until the design of the software is clearly laid out from a user's perspective. Usability and agile are well compatible and they can work together.

The future work will continue to the present work on more detail level using qualitative analysis by observing specific agile methods and usability practices that can be used to bridge the gap between the agile development and usability for developing usable software systems.

ACKNOWLEDGMENT

First, I would like to thanks ALMIGHT ALLAH for giving me the power and will to complete the study. I would like to thank Professor Khalid Khan all the help he offered in the course of my studies. His encouragement and guidance have been of great value to me Personally, I would also like to thank my parents and my partner for their support in my studies.

REFERENCES

[1] J. Schneider and R. Vasa, "Agile practices in software development – Experiences from Student Projects," Proc. 2006 Australian Software Engineering Conference (ASWEC'06), ACM Press, 2006, pp. 401-410.

[2] J. Nielson, "Usability engineering lifecycle," IEEE computer vol. 25, issue 3, 1992, pp. 12-22.

[3] "What is agile software development," May 14, 2006. [Online] Available: http://www.agilealliance.org/show/2 [Accessed: February 2010]

[4] J. Koskela, "Software configuration management in agile methods," VTT publication 514, ESPOO, 2003, pp 1-54.

[5] H. Keramati and S. H. M. Hosseinabadi, "Integrating software development activities with agile methodologies," Proc. 6th ACS/IEEE International conference on Computer System and Application (AICC SA '08), 2008, pp. 749-754.

[6] P. McBreen, "Quality assurance and testing in agile projects," McBreen Consulting, [online] Available: http://www.mcbreen.ab.ca/talks/CAMUG.pdf [Accessed: December 2009]

[7] M. Huo, J. Verner, L. Zhu, and M. A. Babar, "Software quality and Agile Methods," Proc. 28th Annual International Computer Software and Applications Conference (COMPSAC'04), 2004, pp. 520-525.

[8] J. C. Lee and D. S. McCrickard, "Towards extreme(ly) usable software: exploring tensions between usability and agile software development," Proc. AGILE 2007 conference, (Agile '07), IEEE Press, 2007, pp. 59-71.

[9] A. Navalkar, "Usability Engineering – Quality Approach, " (ISO 13407), ISO Process Consultant, [Online] Available: http://www.humanfactors.com/downloads/documents/UsabilityISO.pdf [Accessed: Nov 2009]

[10] "What is usability engineering," April 5, 2001. [Online] Available: http://www.sun.com/usability/about.html [Accessed: December 2009]

[11] "Usability basics," [Online] Available: http://usability.gov/basics/index.html [Accessed: December 2009]

[12] K. Vredenburg, S. Isensee and C. Righi, user-centered design: An integrated approach, Upper Saddle River, NJ: Prentice Hall PTR, 2002

[13] "What is user centered design," UPA, [Online] Available: http://www.upassoc.org/usability_resources/about_usability/what_is_ucd.html [Accessed Jan 2010]

[14] D. Kane, "Finding a Place for discount usability engineering in agile development: Throwing down the gauntlet," Proc. Agile Development Conference (ADC'03), IEEE Press, 2003, pp.40-46.

[15] D. Fox, J. Sillito and F. Maurer, "Agile methods and user-centered design: How These Two methodologies are being successfully integrated in industry," Proc. AGILE 2008 conference (Agile '08), IEEE Press, 2008, pp. 63-72.

[16] D. Sy, "Adapting usability investigations for agile user-centered design," Journal of Usability Studies, vol. 2, Issue 3, 2007, pp. 112-132.

[17] J. Ungar, and J. A White, "Agile user centered design: Enter the design studio," Case Studies in Proc. Conference on Human factors in computing systems (CHI '2008), ACM press, 2008, pp. 2167-2177.

[18] M. Najafi and L. Toyoshiba, "Two case studies of user experience design and agile development," Proc. AGILE 2008 conference (Agile '08), IEEE Press, 2008, pp. 2167-2177.

[19] S. Chamberlain, H. Sharp, and N. Maiden, "Towards a framework for integrating agile development and user-centered design," Proc of 7th International conference on XP and agile process in software engineering (XP '06), Springer-Verlag Berlin Heidelberg, 2006, pp. 143-153

[20] J. Eklund and C. Levingston, "Usability in agile development," UX research, 2008, [online] Available: http://www.uxresearch.com.au/wp-content/uploads/2008/10/agile.pdf [Accessed: March 2010].

[21] M. Düchting, K. Nebe and D. Zimmermann, "Incorporating user centered requirement engineering into agile software development," Human-Computer Interaction, Part I, HCII , Springer-Verlag Berlin Heidelberg 2007, pp. 58–67.

[22] T. Memmel, F. Gundelsweiler and H. Reiterer, "Agile human-centered software engineering," Proc. HCI 2007, British computer society Press, 2007, pp. 167- 175.

[23] H. Sharp, H. Robinson, and J. Segal, "Integrating user centered design and software engineering: a role for extreme programming," [Online]Available:http://www.ics.heacademy.ac.uk/events/presentations/376_hcie-sharp2.pdf [Accessed: Dec 2009].

[24] J. Kollmann, H. Sharp and A. Blandford, "The importance of Identity and vision to user experience designers on agile projects," Proc. AGILE 2009 conference (Agile '2009), IEEE Press, 2009 , pp. 11-18

[25] J. Patton, "Hitting the Target: Adding interaction design to agile software development," Proc. Conference on Object Oriented Programming Systems Languages and Application (OOPSLA 2002), ACM press, 2002, pp. 1-7

[26] J. C. Lee, D. Scott, McCrickard, K. T. Stevens, "Examining the foundations of agile usability with eXtreme scenario based design," Proc. AGILE 2009 conference (Agile '2009), IEEE Press, 2009, pp. 3-10

[27] J. Ferreira, J. Nobel and R. Biddle, "Agile development iterations and UI design," Proc. AGILE 2007 conference (Agile '2007), IEEE Press, 2007, pp. 50-58

[28] J. Ferreira, J. Noble and R. Biddle, "Up-Front Interaction Design in Agile Development," Proc. XP 2007, Springer-Verlag Berlin Heidelberg, 2007, pp. 9-16.

[29] J. C. Lee, "Embracing agile development of usable software systems," Doctoral consortium paper in Proc. Conference on Human factors in computing systems (CHI '06), ACM Press, 2006, pp. 1767-1770.

[30] M. McNeill, "User centered design in agile application development," [online]Available:http://www.thoughtworks.com/pdfs/agile_and_UCD_MM.pdf [Accessed: November 2009]

[31] H. Obendorf and M. Finck, "Scenario-based usability engineering techniques in agile development processes," Case study in Proc. Conference on Human factors in computing systems (CHI '08), ACM Press, 2008, pp. 2159-2166.

[32] Z. Hussain, P. Wolkerstorfer, M. Tscheligi, M. Lechner, S. Shahzad, R. Sefelin and H. Milchrahm, "Probing an agile usability process," Proc. Conference on Human factors in computing systems (CHI '08), ACM Press, 2008, pp. 2151- 2157.

[33] A. Holzinger, M. Errath, G. Searle, B. Thurnher and W. Slany, "From extreme programming and usability engineering to extreme usability in software engineering education (XP+UE->XU)," Proc. 29th Annual International Computer Software and Applications Conference (COMPSAC'05), IEEE Press, 2005, pp. 169-172.

[34] M. Singh, "U-SCRUM: An agile methodology for promoting usability," Proc. AGILE 2009 conference (Agile '09), IEEE Press, 2009, pp. 555-560.

[35] P. McInerney and F. Maurer, "UCD in agile projects: dream team or odd couple?" Interactions, vol. 12, issue 6, ACM Press, 2007, pp. 19-23.