International Journal of Software Engineering & Computer Sciences (IJSECS)
ISSN: 2289-8522 , Volume 1, pp. 29-40, February 2015
©Universiti Malaysia Pahang
DOI: http://dx.doi.org/10.15282/ijsecs.1.2015.3.0003

USABILITY EVALUATION METHOD FOR AGILE SOFTWARE DEVELOPMENT

Saad Masood Butt[1] and Shahid Masood Butt[2]

[1]Faculty of Computer Engineering, Bahria University Islamabad, Pakistan,
[2]Hamdard Institute of Management Sciences, Hamdard University Islamabad, Pakistan
Email: saadmasoodbutt668@yahoo.com

ABSTRACT

Agile methods are the best fit for tremendously growing software industry due to its flexible and dynamic nature. But the software developed using agile methods do meet the usability standards? To answer this question we can see that majority of agile software development projects currently involve interactive user interface designs, which can only be possible by following User Centered Design (UCD) in agile methods. The question here is, how to integrate UCD with agile models. Both Agile models and UCD are iterative in nature but agile models focus on coding and development of software; whereas, UCD focuses on user interface of the software. Similarly, both of them have testing features where the agile model involves automated tested code while UCD involves an expert or a user to test the user interface. In this paper, a new agile usability model is proposed and the evaluation is of the proposed model is presented by practically implementing it in three real life projects. . Key results from these projects clearly show: the proposed agile model incorporates usability evaluation methods, improves the relationship between usability experts to work with agile software experts; in addition, allows agile developers to incorporate the result from UCD into subsequent interactions.

Keywords: Agile software development; User centered design; User interface; Usability evaluation.

INTRODUCTION

In today's age, the software development process has turned dynamic and flexible to achieve better software quality. The ongoing major research using current agile methods aims to achieve usability of the software. Consistent gaining popularity of the agile method is due to the iterative and fast development in fewer spans of time. This agile feature of rapid development has given a big success in the domain of software models, but a big question arises on how to make software usable and increase its usability in less time. With the passage of time, a new field arises in the field of computer science called Human Computer Interaction (HCI). One of the common topics in HCI is User Centred Design (UCD). UCD design website, mobile applications etc. are based on the user experience. Main goal is to design an interface with simplicity and efficiency to achieve the user's goal. Still, the question is how to incorporate the UCD in Agile model? Both agile and UCD have iterative nature. While agile focuses on working software exclusive on the coding and development of software; whereas, UCD focused on interface designing and making interactive interface design for the users.

Similarly, both agile and UCD have testing methods. Agile model follows automated code testing; while, UCD have usability experts or user for the testing of software. Many software companies focus on usability in agile methods but UCD practices are still behind. The association of usability approaches in agile methodology is a promising combination (Mcinerney and Maurer, 2007). However, agile and usability specialists possess diverse expertise and knowledge. Hence, the significant issue faced is that agile development methods merely focus on workable software along and explicitly ignorance of usability issues. Along with this, many agile methods do not engage end users, which is a serious draw back in the development phase. Using usability approaches require real users which is a challenge. In this paper, we propose a new software model proposed, based on literature. Next section shows explains on how UCD can be integrated with an agile model. In the main section, some projects are presented with the results and in the end present the conclusions.

BACKGROUND

Conventional approaches in design and implementation of software models have their own draw back. Most of these methodologies have different challenges like: continuous change in software requirements and other schedule delays which overall slows down the end product in software development. Following agile methodologies along with the reasons on product iteration and reduction in team coordination cost result sometimes to unproductivity.

Usability in agile software has become an important perspective for software quality assurance. Awareness of actual end users in software development life cycle is mostly a critical aspect. Most of the time, word customer is used instead of end users. Customers are giving the requirements, taking part in software design and are also involved in testing of software from the first stage of software development till the end. So the real query behind is this: Are customers the actual end users or just representatives of end users in software development? Conventional software design methodologies have its darker side and that is taking more time in developing and meeting the requirement in a given deadline. These drawbacks generate the need of new strategy like software agile development. Agile development overcomes the time barrier and software completion within a given framework along with frequent customer involvement and capability to handle requirement changes if any (Agilealliance, 2013).

Agile methodologies focus on incremental and iterative approaches that not only supports overall optimization in development process but also support developers to generate product incremental results in small amount of time (Keramati and Hosseinabadi , 2008). Agile methodology targets in delivering a piece of the working code in a small frame of time. The iterative approach can fabricate good quality code, but it does not assure consistent user interface because the focus is only in delivering small pieces of functionality after every interaction (Kollmann et al., 2009). This establishes the fact that agile methods can efficiently produce working software in limited time but cannot make sure the delivery of usable software. Hence, to generate usable interactive software, UCD is a must in agile methodologies (Patton, 2002). Gigantic companies like IBM put emphasis on the concept of usability in software development. This quality attributes not only incrassates the market value of that product but also makes it reusable (Shackel, 2007). Similarly, Microsoft focused on the benefits of usability in software development, in a recently published report (Microsoft, 2013). The reason to achieve software usability is to help minimize the training cost on usable software and enhance the user acceptance of the software by the end users.

McBreen (Koskela, 2003) elaborated the notion of agile software development methods as a strategy of rapid development such that spending less time on analysis and design. . Customer's feedback makes the development process speedy by following an incremental and iterative approach. However, the authors stated that software interface designing cannot be supported due to the incremental and iterative way of agile software development, which is obligatory for the development of usable and interactive software (Lee and McCrickard, 2007).

Kane (Kane , 2003) argued that the disintegration of usability in agile methods Also, integration of usability in agile methods will boost overall product usability along with the satisfaction of the end user.

Fox et al (Fox et al., 2008) addressed that no agile methods can guarantee the usability of software or product. Usability techniques in development methodology are usable in the development of interactive and usable software design especially in case of agile software methodologies. Though, agile methods are widely in use in software industry yet the software usability is an objectionable issue. Therefore, it seems necessary to integrate usability to the agile way of developing software (Sy, 2007). Yet the contribution from authentic end-users is still another question (Sohaib and Khan, 2010). Like in Extreme Programming (XP) the customer's onsite presence can work but uncertainty remains there that whether they are the actual users or representative of end users (Najafi and Toyoshiba, 2008). Therefore, XP might deliver good quality software but fails to satisfy the customer.

Sharp et al (2013) intended to focus on customers and Extreme Programming (XP) collaboration and to emphasize on the effectiveness of UCD in XP. The paper focused on the absence of reliance between customers and developers. The findings stated that the involvement of the right user not only helps in getting the right requirements done but also ensures the usability of the software. However, it is an issue to differentiate customers and real users in software development (Memmel et al., 2007). Memmel et al (2007) introduced a model called Cross-discipline User Interface design and Software Engineering (CRUISER) lifecycle, which combines HCI with SE. Involvement of the user in the designing stage produces usable results, that can optimize the development process by (x y z how it does that please mention here and how it is helpful).

Patton (2002) focused on the ten facts that can help agile specialist to integrate UCD during the software development but was unclear about its implementation. Lee et al (2009) proposed a new model called Extreme Scenario-based Design (XSBD). This model emphasized on the concept of central design record (CDR) that helps to maintain the communication between: agile experts, stakeholders and usability specialists. The model helps the usability expert in designing issues and conducting usability testing.

Ferreia et al (2007) argued about introducing Big Design Up Front (BDUF) in agile development methodologies. To avoid later changes in design, BDUF generates design details before programming modules and testing. However, the companies using agile methods go otherwise and spend a lot of time on code testing in order to build up workable software. These techniques test every software module in diverse environment but none of these methodologies directly support usability testing and usability evaluation of software Agile specialists spend no time on extensive usability testing with the end users in order to get product usability (Sohaib and Khan, 2010).

Singh (2009) proposed a new methodology called U-SCRUM which supports two kinds of product owners: product functionality and product usability. Unlike a conventional SCRUM models, which focuses in development of software that is workable in a small amount of time? Hence, following U-SCRUM tactics can ensure usability in all SCRUM projects. McNeill (2013) described that techniques of storyboarding is an iterative approach and a low reliability prototyping in agile that can perk up the software design instead of spending time on code modules. Hence, enhanced versions of these products can develop a complete focus on end-users by adding UCD in agile approaches. High budget and long-time periods both are not necessary for the Usability testing's efficiency. Techniques such as discount usability allow engineers to work in team by thinking distinctly. Other methodologies: scenario-based, walkthroughs, card sorting and heuristics make the process much cheaper, vibrant and easy. These techniques can be applied at an earlier phase in the life cycle and the implementation phase so that any major or minor usability issues can be notified. The working framework mentioned (Sy, 2007), discount usability model is implemented within agile iterative settings and becomes more effective.

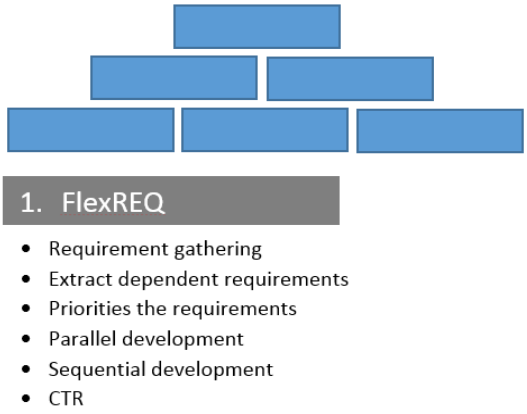
Agile development model implements an iterative approach and has a very stern time frame wherein daily scrum based meeting is held to update the team with the recent changes happening in the project scope. Researchers came up with possibility of integration of usability methods and agile model. Adopting agile approaches while focusing on usability-centred design leads to an amazing experience that will result in a timely delivery of a highly usable end product. The above discussion it makes clear that the best way to evaluate the software design is to understand the end users and their interaction with the system. To ensure the software usability and quality, UCD is the need of the hour. Hence, for the development of usable software, the agile developer needs to include interactive design and good usability methods in their development methodology (NN, 2013).

PROPOSED AGILE USABLITY MODEL

The Agile Software Development methods are flexible and dynamic .The main focus of agile software development methods is on accomplishing functional requirements and deliver the in running software at the end of each iteration. However, the failures can occur in running software due to due to its lack of usability. In this case, revising the software features for tracing back the fault cost a lot and reduces the quality of software. Therefore, to address the failures of software due to the lack of usability we proposed a software development model named as Agile Usability Software Model. This model incorporates UCD and Usability in the agile software development process. The proposed model focuses on the fact that UCD and usability issues need to be fixed before (or with) the software development. In addition, to the previously used typical testing like acceptance testing or unit testing, usability testing is considered as a part of the process. In this way, agile software developers and usability experts can work together to develop interactive software. The proposed model comprises of five processes such as: FlexREQ, Interactive User Interface Designing, Cognitive analysis of software interfaces, Development & Testing and Deployment, described below.

FlexREQ PROCESS

The first process is FlexREQ (Masood and Ahmad , 2012) as shown in Figure 1. In the traditional software development, eighty percent effort is consumed in SRS (Software Requirement Speciation) document development. Customers contribute in this process of producing SRS document. However, sometimes when the customer is a business representative from a non-technical background and does not understand the highly technical SRS document, it can affect the final software. Unlike, traditional software development methods agile method has less or no documentation and thus the role of SRS is eliminated. The iterative approach of agile methods helps in adding the missing functions to the next release which went missing in the prior one; while incremental approach helps to release the software in a form of small modules later merging in one big modules for final deployment at the end.



1. FlexREQ

- Requirement gathering
- Extract dependent requirements
- Priorities the requirements
- Parallel development
- Sequential development
- CTR

Figure 1. The FlexREQ

FlexREQ is an automated requirement elicitation process that helps project team leader to manage the customer requirements and select the few requirements based on customer interest for the first release. Requirements selected for first release can undergo either parallel development or sequential development. For every release it calculates the CTR (Cost, Time and Resources) and records the process. This FlexREQ process helps to collect user requirements and manage them efficiently in a small amount of time.

INTERACTIVE USER INTERFACE DESIGNING (IUID) PROCESS

To develop interactive software, (like video games, mobile applications or operating system like Windows 8) high level of Usability tested by usability experts is deemed. It has become a challenge for agile experts to focus on UCD and usability for the development of Interactive software. The second process of the proposed model is the IUID (Interactive User Interface Designing) shown in Figure 2 that is designing of interactive interface mock-ups. IUID makes an objective to design the interface that is simple and interactive so that users can achieve its goal.

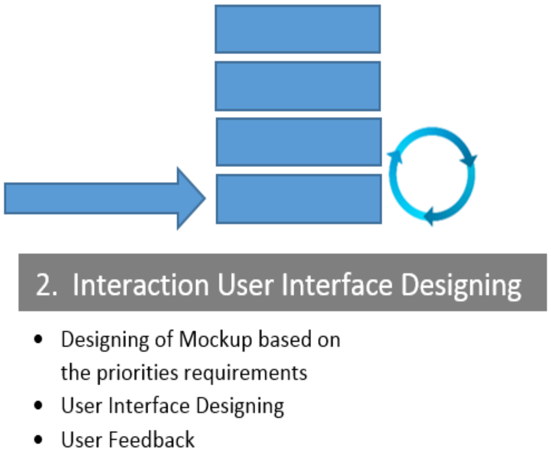

2. Interaction User Interface Designing
Designing of Mockup based on the priorities requirements
User Interface Designing
User Feedback

Figure 2. IUID Process

COGNITIVE ANALYSIS OF SOFTWARE INTERFACES (CASI) PROCESS

CASI (Masood and Ahmad, 2012) as shown in Figure 3, is a usability evaluation process that evaluate every mock-up for the usability before the coding starts. The main objective of this process is to find Usability bug in the mock up and fix those bugs.

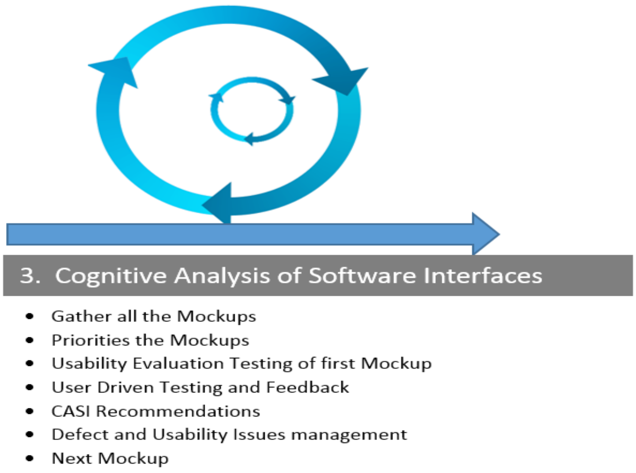

3. Cognitive Analysis of Software Interfaces
Gather all the Mockups
Priorities the Mockups
Usability Evaluation Testing of first Mockup
User Driven Testing and Feedback
CASI Recommendations
Defect and Usability Issues management
Next Mockup

Figure 3. CASI Process

Mock-ups having poor usability and bugs are sent back to the previous iteration to get a fix. Successful mock-ups from CASI will move to the coding phase.

DEVELOPMENT & TESTING PROCESS

As shown in figure 5, successful mock-ups from CASI process will go for coding and testing. All mock-ups will be again tested one by one after coding. This involves Unit and Acceptance tests. The difference between the post-code (unit and acceptance testing) and pre-code (CASI) is that unit and acceptance testing will test the functional requirement of software whereas CASI will test the usability of the software.

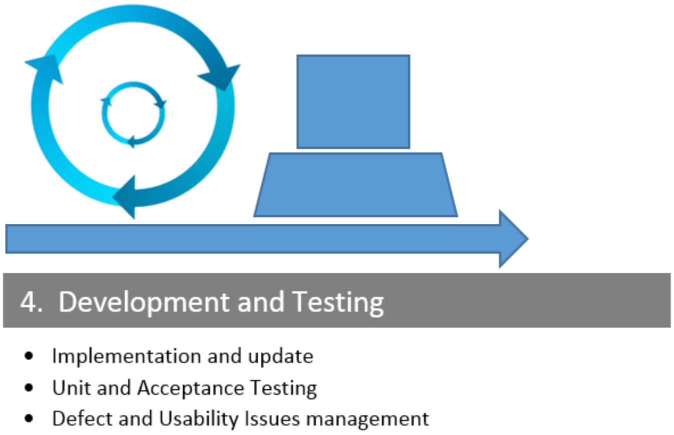

4. Development and Testing
Implementation and update
Unit and Acceptance Testing
Defect and Usability Issues management

Figure 4. Development and Testing Process

DEPLOYMENT PROCESS

After all of the iterations, the software is now ready for deployment at the customer site as shown in Figure 5. The proposed model has this much capacity to accept new requirements or modify the existing requirements based on the customer request.

5. Deployment
Installation
User Feedback

Figure 5. Deployment Process

PROJECT PROFILES

The proposed model has been practically tested in various companies proves its applicability on the diverse software projects. To assess the performance of this model, real world projects were chosen and each project was unique in implementation and business domain. For example, some projects need to develop new software while others were to release the new version of the software. Other projects were to improve the Usability of the existing software. This section highlights the actual practice of proposed agile usability software model.

PROJECT 1:

The first project, factiously named P1, develops and markets civil engineering software for new students enrolled in civil engineering discipline. it is a software that helps civil engineering students to construct and present Town Plan in 3D view. P1 was developed based on the proposed model with a team of five software engineers, one domain expert and one UI designer, one usability expert, one coder and one tester. The project was divided into two or three iterations and each iteration lasted five days.

P1 started with the collection of requirements from the domain experts like faculty member and students. During the first meeting, the requirements were prioritized based on user stories called as wish-list (or feature list). These wish-list of requirements was recorded using FlexREQ. Every independent wish-list has its own time schedule. During pre-release planning meeting product's release date was recorded. Wish-list selected from FlexREQ was ready for IUID. This involved domain experts to work with UI designers to design mock-ups using Balsamiq [ref]. This helps in designing simple and interactive mock-up for every wish-list. Missing features in mock-up were added in the next iteration. All of the mock-ups were then ready for Usability Evaluation. The domain expert did that and results were then recorded. Poor mock-ups having usability bugs were send back to IUID to fix these bugs. Within the first iteration the successful mock-ups were coded and tested by the developers. Unit testing and acceptance testing were performed including the Usability expert, which was also tested from the perspective of Usability and UI. The final module was deployed to the customer. Customer feedback was noted or usability issues were highlighted to fix in the next iteration. On the end of all iterations, all modules are integrated and the complete software was deployed to the customer. Feedback regarding changes or omission or usability issues were provide to the developers and usability expert. After these changes the software will be tested by UI expert and client.

PROJECT 2:

The second project, P2, is developed software. Now, the client wants to release the next version of software because the previous have poor UI and have poor usability. The new version of software is developed using the proposed model. P2 involved the team consist of UI designer, Usability expert, Coder and onsite customer. P2 involved the release of the next version of existing software. UI designer have a meeting with customer where the new requirements and changes in an existing requirement are noted. All this will be added in FlexREQ to get the simple wish-list of requirements. So UI designers did not perform any change in overall design of the software but they just need to redesign the existing UI design with new features or changes in the existing features. New mock-ups are developed and now ready for usability evaluation. The onside customer does this before these mock-ups are sent to the development team. Usability issues will be identified by CASI and the report will be sent back to UI designer to fix the design issues. After fixing the design issue the development is now ready to code the requirements in the wish-list. These changes are small but the development team takes minimum time to develop. Finally, they will be sent for acceptance testing. During the testing the customer walked through the new UI, usability is evaluated and how the new software is working.

PROJECT 3:

The third project P3 is already developed but the users of the software had difficulty in understanding the various functions and had undertaken task difficulty. The project is then evaluated for its Usability using CASI. It was discovered that the software was developed by hiring developers from freelancer website like (ODESK). P3 consist of UI designer, Usability Expert, Developer and User. In P3, the interface of software is selected and loaded in CASI for the evaluation of Usability; where those having poor usability are send to UI designers. UI designer discussed with users in order to improve the design according to user needs. Final mock-up will be re-evaluated for the usability and later send to the developer for final development. Some interfaces, which the user wants to revise, took considerable amount of time because other functions are dependent on that interface. At that point UI designer had meetings with developers to redesign the interface without affecting other functions

RESULTS

In this section, some results are presented based on the project. The results also highlight the importance of usability in agile methods where both usability experts and agile software development practitioners can work together to produce quality software.

ITERATIONS SEEM TO IMPROVE UI DESIGN

The first result observed from the projects is that iteration helps user interface designer to improve the design. This iterative nature is very cost effective that helps to cover the interface issues before the development of the software.

ITERATIONS SEEM TO SUPPORT USABILITY EVALUATION

Usability evaluation means how easy for the user to work with the real software to achieve its goal or to what extent the software is easy to use. Usability evaluation is a well-established discipline of HCI. The iterative nature of agile software model seems very helpful to evaluate the usability of software at a very early stage. This helps in fixation of usability bugs during the design stage and gets proper feedback from the user before the coding starts. Moreover, usability evaluation seems fit with the agile model as it seems with the acceptance testing.

CASI RESULTS CHANGES IN DEVELOPMENT

Testing in agile model helps to identify the bugs, which is then improved in the next iteration. This is the reason why iteration process is universally accepted for software development (Masood and Ahmad, 2012). Similarly, the CASI process helps to identify usability bugs in the user interface which is fixed in the next iteration of software release. Having iterations in agile model is seen as an opportunity for usability evaluation to identify usability bugs or can accommodate any new feature; so that the next iteration is seen as an opportunity to fix the usability bugs or accommodate those changes.

PROPOSED MODEL HELPS DEVELOPS AND DESIGNERS WORK TOGETHER

Final result identified in the proposed model and project is a better relation between UI designers and software developers. It is obvious from the proposed model that development of interactive software can accommodate both UI designers and agile software developer under one model. Table 1 gives a comparison of the proposed model and the conventional agile software development process.

Table 1. Comparison of software development activities through proposed model versus agile software development methods

	Proposed Model	**Agile Models**
Requirement Gathering	FlexREQ Technique [24]	Manual way of writing User Stories [26]
Designing	IUID and Users	Software designers and Developers
Usability	CASI	Not Defined[6][7][8]
Development	UI Designer, Usability Expert, User and Coder	Software developers
Testing	CASI, Acceptance and Unit Testing	Unit test, Acceptance Test [25]
Deployment	Get User Feedback	Not Defined
Aim	Develop Software up to User Expectation and focus Usability	Working Software [12]

CONCLUSION

This paper shows the importance of resolving usability issues in agile methods.. A model has been proposed to incorporate usability in agile software development. The proposed model was deployed in three real life projects to show how UI, Usability Evaluation fit into the structure of agile software development t. These projects discussed that it took considerable time to design the interactive interface and maintain a good connection between Users, UI designers, Usability Experts and developers. The results revealed that Iteration improves UI design, Iteration supports UE, CASI result changes in development and proposed model in order to accommodate Software Developers and UI Designers.

This point reflects a coherent picture of putting UI and Usability Evaluation together in agile software development processes for better quality and useable software. Its iterative nature allows UI designer and Usability evaluation to identify bugs and fix the bugs in the next iteration, which are not expected but happens and advantages are seen.

REFERENCES

Butt, S. M., & Ahmad, W. F. W. (2012). Analysis and Evaluation of Behavior in Software Interfaces using an Expert System. International Journal.

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, ppAO-46.

D, Fox, 1. 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.

D Sy, "Adapting usability investigations for agile user-centered design," Journal of Usability Studies, vol. 2, Issue 3. 2007, pp, 1 12-132.

Ferreira, 1. Noble and R. Biddle, "Up-Front Interaction Design in Agile Development," Proc, XP 2007, Springer-Verlag Berlin Heidelberg, 2007, pp, 9- 16.

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

H. Sharp, H. Robinson, and 1. Segal, "Integrating user centered design and software engineering: a role for extreme programming," [Online]A vailable:http://www.ics.heacademy.ac. uk/events/presentationsl376_hcie sharp2,pdf [Retrieved: December 2013].

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.

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, 1 1- 18.

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

Koskela, "Software configuration management in agile methods,"VTT publication 5 14, ESPOO, 2003, pp I-54.

Lee, J. C., & McCrickard, D. S. (2007, August). Towards extreme (ly) usable software: Exploring tensions between usability and agile software development. In Agile Conference (AGILE), 2007 (pp. 59-71). IEEE.

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, 2 167-2177.

Masood Butt, S., & Ahmad, W. F. W. (2012, June). Handling requirements using FlexREQ model. In Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on (pp. 661-664). IEEE.

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

Microsoft Corporation: “Usability in Software Design”, (Online) Available http://msdn.microsoft.com/en-us/library/ms997577.aspx. (Retrieved: December 2013).

M McNeill, "User centered design in agile application development," [online]Available:http://www.thoughtworks.com/pdfs/agile_and_UCD MM, pdf [Retrieved: December 2013].

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

NN, “Agile Development Projects and Usability”, (Online) Available: http://www.nngroup.com/articles/agile-development-and-usability/ (Retrieved: December 2013).

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.

Parsons, D., Lal, R., Ryu, H., & Lange, M. (2007). Software development methodologies, agile development and usability engineering. ACIS 2007 Proceedings, 21.

Shackel, B. (2007). IBM makes usability as important as functionality. The Computer Journal, 29(5), 475-476.

Sohaib, O., & Khan, K. (2010, June). Integrating usability engineering and agile software development: A literature review. In Computer Design and Applications (ICCDA), 2010 International Conference on (Vol. 2, pp. V2-32). IEEE.

"What is agile software development," May 14, 2009, (Online) Available: http://www.agilealliance.org/show/2. Retrieved on December 2013.