Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2011) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

WCIT 2011

Usability studies in agile software development

Deniz İren [a] *, Mustafa Dalcı [b], Banu Aysolmaz [c]

[a]METU Computer Center, Ankara 06800,Turkey
[b]METU Computer Center, Ankara 06800,Turkey
[c]METU Informatics Institute, Ankara 06800,Turkey

Abstract

Agile software development approaches are being used increasingly by practitioners since the announcement of Agile Manifesto over a decade ago. Lately, projects adopting agile software development approaches are encountered in public sector as well. In METU Computer Center, we have initiated a number of software development projects using agile methods recently and reorganized our Project Management Office (PMO) to enable the PMO to support agile projects. Research indicates that, usability studies, which aim at identification and elimination of causes of inefficient and ineffective usage of IT products and services, have significant impact on project success. METU Computer Center provides usability testing services to researchers and practitioners with the HCI Lab since its establishment in 2006 as a result of a TUBITAK-BAP project. This research includes all effort spent in combining the best practices and our experiences in both areas; agile software development and usability studies, along with an extensive literature study. This article aims at presenting the results of our studies, a review of current literature and best practices about usability studies undertaken within agile projects as well as discussing these practices in the light of our own experiences on the subject.

Keywords: Agile software development, usability;

1. Introduction

Adoption of agile software development approaches are spreading ever since the publication of Agile Manifesto ten years ago [1], [2], [3], [4]. Lately, projects using agile development approaches are encountered in Turkish public sector as well as the private software developing organizations. A short time ago a managerial decision was made in METU Computer Center to support utilization of agile software development approaches in organizational projects. Then the Project Management Office reorganized itself to develop the capability to support agile projects.

A definition of usability is given by ISO/IEC 9241 standard as the degree of effective, efficient usage of a product by a predefined group of people for defined purposes [5]. The Human Computer Interaction Lab, which is active since its establishment as a result of a TUBİTAK Scientific Research Project in 2006 in METU Computer Center, is being used for conducting usability studies for both academic and industrial purposes.

In this article we present the results of our study aiming at integrating two practices; agile development and usability studies, which we are perform separately in our organization. The first section of the article is an introduction to the setting of the research environment. Section 2 describes the implementation of usability practices in agile projects. Finally, section 3 discusses the usage of agile software development approaches and usability tests

* "Deniz İren (*)". Tel.: +90-532-583-3389
E-mail address: diren@metu.edu.tr

in concert, and provides vision for future work to be done.

2. Usability in Agile Project

The nature of agile projects provides early value delivery to the customer and achievement of fast results with continuous integration. Because of these reasons the software is developed iteratively and it is assured that the existing product meets the definition of "done" at every iteration by means of tests. This indicates that the tests are an essential part of every agile development iteration. As part of these tests and other development activities, usability practices should also be handled at every iteration; and a common understanding of usability activities must be reached at the overall organization.

Satisfying common needs of projects carried out in the same organization such as ensuring usage of the same terminology, supporting the utilization of industrial best practices and standards is the responsibility of the Project Management Office (PMO). Similar to PMO skills, usability is also a skill required among the projects. The utilization of usability expertise and usability practices among the projects must be carefully planned in the organization. The next sections discuss the possible utilization of usability expertise in the organization and the agile practices that needs to be integrated or in relation with usability concepts.

2.1. Locating Agile Expertise in the Organization

As usability expertise is not yet a common role, it is not easy to find usability professionalism for each project in the organization. Yet, common models we encounter to meet usability requirements of the projects can be listed as follows:

- Team members any of which is not expert share responsibility of usability activities
- One or more usability expert exists in the organization but shared among projects
- The usability expert is assigned as a dedicated team member in each of the projects
- A consultant or expert outside of the organization provides service when necessary

As discussed, it is important to reach a consensus on usability concepts among project teams in an organization. Developing a standard usability expertise inside the organization to be used among projects results in compatible end products from the user experience perspective and a certain characteristic feel which reflects the organizational identity better. Moreover, in agile projects which are being performed by small, surgical teams, usability expertise can be hard to find. Thus, usability expert should be seen as a customer. Acquiring this skill from outside the project team may be more feasible and cost effective [6].

To uncover usability expertise allocation in agile projects, we conducted a survey on how organizations place usability expertise among agile projects in the organization. 90 participants took part all of which has usability requirements in their projects. The results show that 64% of the organizations utilize usability experts, while only %36 distributes the responsibility among the team members who are not experts. %60 of the organizations with a usability experts share the expertise among projects, while %36 employ agile team-specific experts, and only 3.5% utilize external sources.

The results of this survey shows us that though the benefits can change according to organizational characteristics like size of the organization, agile team, usability requirements of the projects; organizations tend to share usability expertise among agile project teams. In this way, both the costs are cut down and standardization is achieved in usability practices of small agile teams.

2.2. Agile usability practices

2.2.1. Test first approaches

Test First approaches, namely Test Driven Development and Test first Design are best practices which are being strongly suggested by many agile software development models. These approaches propose identification and authoring the tests prior to the development of software code, in order to completely satisfy the requirements. Similarly, usability tests should be applied before even the user interface design process. This approach is not appropriate to be used with the waterfall-like sequential lifecycle models which mandate processes resulting in testing as late as possible.

2.2.2. Early Feedback

Continuous and early feedback is one of the essential values of agile software development methodologies [2]. Organizations cannot avoid feedback. Project teams which fail to listen to their customers and users throughout the development process are facing the risk of getting the feedback in terms of bad comments, low profits and terminated contracts when the software development is over. Since it is not possible to avoid the feedback, it needs to be managed [7]. A significant portion of software development lifecycle costs is related with the effort spent for reconfiguration and revision. The most important causes of rework are non-compliance to the system requirements, failure to identify user requests upfront and usability problems [8]. Thus it is advisable for the business representatives work on site continuously, together with the developers. The feedback from potential users and usability experts should be used in a way that forms a basis for the design [7]. In order to achieve real agility in software development and minimizing the rework and reconfiguration, usability tests should be conducted not only aiming at describing the problem but also prescribing the solution. RITE method, which is in use for a long time, accepts this approach as the transition of verification process into design process and proposes conducting a continuous series of test-fix-test-fix activities until the product becomes acceptable [9]. In a similar perspective it is advised to a conduct “a morning of usability testing per iteration” rather than “a morning of usability test per month” in order to align the guerilla usability testing with agile software development [10].

2.2.3. Iterations

Agile software development projects consist of small, time-boxed iterations. The iterations contain analysis, design, implementation and test phases within. At the end of each iteration a tested, integrated, potentially releasable product fragment is delivered. Thus the delivered part of the product is assumed to be usability tested and achieved a desired usability level. The scope of the iteration is determined by selecting the user stories in the beginning of the iteration. At this point in time the requirements for the user interface and other user interaction methods which will present the system functions to users should be decided. When planning the iteration, it should be kept in mind that two distinct purposes exist for usability studies [11]. These are;

- Prototype development for guiding the design which should be conducted early in the iteration (Pt(n))
- Usability tests aiming at diagnosing usability problems should be carried out at the end of the iteration. (Ut(n))

The non-compliances diagnosed as a result of usability tests should be written down as user stories for fixing in the upcoming iterations (Ufix(n-1)) as shown in Fig 1. The code developed while fixing the diagnosed problems should be tested as soon as possible to cope with the risk of introducing new usability non-compliances to the system [12].

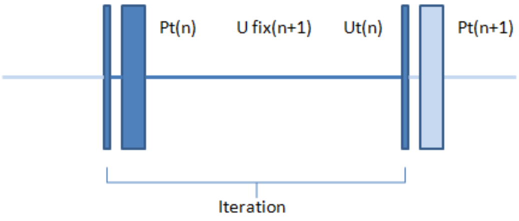

Fig. 1.Timing of usability studies within agile development

2.2.4. Software Engineering Practices

Usability requirements should also be captured during the requirements analysis and specification phases of an agile software development project. These usability requirements can be recorded as *usability stories*. Usability stories reflect the intended usage expression of the usability definition of ISO 9241 [5]. The predefined user expression, which is also emphasized in the usability definition, is covered by personas [13]. Persona analysis is used in elicitation of user stories and enables differentiation of the stories by taking the user specific styles and requirements into consideration. Elicitation and definition of usability stories require different perspectives and capabilities. Users and domain experts often have difficulties in describing the desired user experience. A more complete, consistent and holistic authoring of the user stories is aimed through guidance of usability experts. Prior to the design phase within the iteration, it is possible for domain experts, users and, usability experts, to describe the usability requirements and author the usability stories by using whiteboard or paper prototypes [14]. When applied appropriately, prototypes can be developed quickly and ensured to be detailed enough to constitute a medium for communication [15]. Given adequate effort, prototypes can be detailed in fine grain and become partially reusable in the development of the user interface. The outputs of a prototyping session are used as guidelines in later stages of development, thus, they are used to identify future usability problems in a cost efficient manner.

The developers' observation of domain experts and potential users while they attempt using the parts of the product that are still in the design and development process, provides significant benefits. In doing so, the developers can grasp the users' way of thinking; empathize with the feelings born by the usage of the product and thus make certain that the product is designed to be truly user centric. Even though the entire development team may not be able to participate as observers in the usability studies, by the use of fast reporting techniques, they can discover the problems regarding the user experience and take important steps in user centric design.

2.2.5. Usability Testing Practices

eXtreme Programming (XP), which is one of the first agile software development methods, has stemmed from the idea of two developers working on a single computer during the coding phase [2]. Often a point of discussion ever since its inception, it that this method is successful in arriving at the correct solution, especially in high complexity programming jobs [16]. "Think Aloud Testing", one of the usability test methods, proposes usability expert taking notes while the user uses the product and voices her thoughts. This aims at understanding the user's thought process.

There are findings that this method is often employed in order for the users to confirm the problems already known by the usability expert [17]. Since the purpose of pre-design phase usability studies is understanding the users' thought models rather than determining the problems, a different method can be used. In this method, it is recommended that the users are paired up in front of a whiteboard or a paper prototype, see and experience the first design element and discuss their usage experiences meanwhile and observer documents the ideas that arise during the exercise. Dual participant think aloud testing methods have also been reported to have been used during agile software development projects [18].

2.2.6. Physical Methods That Provide Visualization

In agile software development, story boards are frequently used to assign the user stories to iterations and classify them with respect to their status. Many benefits are identified by visualizing project status and requirements on a physical environment. Among these benefits are continuous visibility of project status, support for communication among team members, identification of work ownership, awareness of other team members' activities, visibility of all changes and foundation of cognitive environment for participatory work [19]. In agile software development projects, it is viable to think that associating the user stories which have functional and user interface requirements with interface and visual elements may bring the same benefits. If there is no user story associated with an interface component on the story board, this is possibly an indication of an unidentified user story. Similarly the functional stories that are not related to any interface component probably indicate a functionality which is factitious or left out-of-scope. Moreover, explaining scenarios to domain experts by reviewing user stories on story boards may be utilized as a validation technique to assure that scenarios are understood correctly by developers [18].

3. Discussion and Future Work

It is obvious that usability studies and agile software development approaches show similarities in characteristics. This indicates a clear harmony. A number of case study results from the literature report that integrating good practices of both approaches by synergic work of usability experts and agile project team members provides improvements to both domains [7]. On the other hand, some researchers argue the benefits by emphasizing the pitfalls. For example, development of GUI-intensive software applications using agile methods results in frequently changing interfaces, which becomes uncomfortable for end users [15].

The results of our survey suggest that there are various ideas regarding from where to get the usability expertise for the projects. We predict that there may be different benefits and pitfalls with different models locating the usability expertise. This area provides interesting research opportunities for us.

We have initiated a case study which employs the good practices described in this article, in order to satisfy the usability requirements of an ongoing agile software development project. As a future work, we are going to analyze the case study findings and develop a generic agile usability methodology in the light of our experiences. Additionally we see potential impact in conducting research to find the optimal source of usability expertise in agile projects.

References

1. «Manifesto for Agile Software Development». Viewed from: www.agilemanifesto.org, 2001
2. K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley Longman Publishing Co., Inc. Boston, 2002.
3. A. Cockburn, Agile Software Development, Addison-Wesley, Boston, 2002.
4. K. Schwaber and M. Beedle, Agile Software Development with SCRUM, Prentice Hall, 2001.
5. ISO/IEC 9241-11, Ergonomic Requirements for Office Work with Visual Display Terminals (VDT)s - Part 11 Guidance on Usability, ISO/IEC 9241-11: 1998 (E), 1998.
6. S. Ambler, Roles on Agile Teams: From Small to Large Teams, viewed from: www.ambysoft.com/essays/agileRoles.html. 18 August 2011
7. L. Miller, Case Study of Customer Input for a Successful Product, Agile Conference, 2005. Proceedings, pg.. 225- 234 (2005).
8. R. S. Pressman, Software Engineering: A Practitioner’s Approach, McGraw-Hill, New York, 2005.
9. M.C. Medlock, D. Wixon, M. Terrano, R. Romero, B. Fulton, Using the RITE Method to Improve Products: A Definition and a Case Study. Usability Professionsals Association 2002, Orlando Florida. (2002).
10. S. Krug, Rocket Surgery Made Easy: The Do-it-Yourself guide to Finding and Fixing Usability Problems. New Riders, 2010.
11. J. Ferreira, J. Noble, R. Biddle , Agile Development Iterations and UI Design, AGILE 2007 ,pg .50-58, (2007).
12. J. Patton , Getting Software RITE, Software, IEEE , nr.25, no.3, pg.20-21, (2008).
13. A. Abran, A. Khelifi, W. Suryn, and A. Seffah, Usability Meanings and Interpretations in ISO Standards, Software Quality Journal, Nr.11, no.4, pg.325–338, (2003).
14. J. Nielsen, Paper Prototyping: Getting User Data before You Code, viewed from: www.useit.com/alertbox/20030414.html. 14 April 2003.
15. T. Memmel, F. Gundelsweiler, H. Reiterer, Agile Human-centered Software Engineering, in Proc. of 21st Brit. HCI Grp. Annual Conference on People and Computers, pg. 167-175.,Swinton, UK, (2007).
16. T. Dyba, E. Arisholm, D.I.K. Sjoberg, J.E. Hannay, F. Shull, Are Two Heads Better than One? On the Effectiveness of Pair Programming, Software, IEEE , nr.24, no.6, pg.12-15, 2007.
17. M. Norgaard, K. Hornbaek, What Do Usability Evaluators do in Practice?: An Explorative Study of Think-Aloud Testing, in Proc. of 6th Conference on Designing Interactive systems (DIS '06). ACM, New York, NY, USA, pg 209-218, (2006).
18. G. Meszaros, J. Aston, Adding Usability Testing to an Agile Project. in Proceedings of the Conference on AGILE 2006 (AGILE '06). IEEE Computer Society, Washington, DC, USA, pg 289-294, (2006).
19. H. Sharp, H. Robinson, M. Petre, The Role of Physical Artefacts in Agile Software Development: Two Complementary Perspectives”, Interacting with Computers, pg. 108-116, (2006).
20. J. Kay, A Test-First View of Usability, Interact. Comput. 21, 5-6 pg, 347-349, (2009).