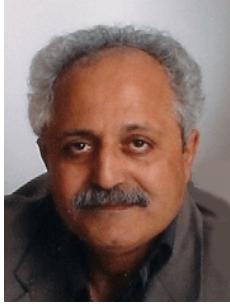


Design and Improve User Experience in Agile Development Cycles

Advantages, Frustrations and More...



Moallem, Abbas

**ISE Department / San Jose State University / One
Washington Square, / San José / CA 95192-0042 USA
E-mail: abbas_moallem@SJSU.com**

ABSTRACT

Transition from a Waterfall development methodology to an Agile software development influences the way that the user experience of products is built and affects the ways that user experience professionals deal with overall ease of use of a product. This article summarizes the advantages and disadvantages of this methodology from a user experience point of view and the lessons learned through a transition from Waterfall methodology to Agile, in an enterprise software applications company.

Keywords

Agile Development, User Experience, Agile UI Design, User Interface Design, Extreme Programming, User-Centered Design

1. INTRODUCTION

Agile software development is a methodological framework to develop software (Agile Manifesto 2001). The Agile methodology in contrast with the Waterfall model of development is iterative and incremental, with an emphasis on building software in a short period of time. Even though the concept is not necessarily new in product design and production (Industrial Democracy and Participative Management Experience, Viklund 1980), this methodology took shape in early 2000 and was largely adopted by many small and big software companies to the extent that the message at the SD Best Practice 2004 conference was "If you're not doing Agile, you're in the past" (quoted by Hamill P. 2004). The advocates and literature of this methodology attribute a variety of advantages to this development framework that include (Agile Manifesto, 2001):

- Highest priority is to satisfy the customer
- Harness change of the customer's competition
- Deliver working software frequently
- More effective team work

- Motivating environment

The Agile methods organize software development into a very small subset of the whole product feature every few weeks or months. The emphasis is on obtaining the smallest workable piece of functionality to deliver business value early, and continually improving it/adding further functionality throughout the life of the project.

To manage the implementation of each feature set or "story", the development teams are organized into "scrums", a process used to manage product development using iterative, incremental practices and dividing development into more or less thirty day iterations each called "sprints". Close monitoring and control are performed through daily scrum meetings or "stand up meetings". After validating the results with the acceptance criteria defined for the requested feature, final work is "Demoed" to all groups.

After analyzing the lessons learned in a "retrospective" meeting, the next stories from product backlogs are evaluated in a "preplanning" and then planned in a "planning" meeting which is a starting point for another Sprint.

The case studies and industry reports are a variety of success and failure stories (Boehm, B., 2002). There are numerous examples of literature reporting advantages and disadvantages of the Agile methodology (Nerur & Balijepally, 2007). There is a whole community of service providers, consultants, trainers, publications, conferences and more that are built around this software development methodology [Agile Alliance, Agile Programmer community Blogs or Agile Development Conference, to just name a few]. This article does not intend to review or evaluate the Agile method in general. What I intend to do is share my experience in the effects of Agile development method on user-centered design methodology and user interface design. I also intend to review the advantages and disadvantages of this development technique on user experience of the product. To my understanding, the body of knowledge on user-centered design in Agile development is still limited and not being evaluated broadly in literature. Most related publications are case studies or anecdotal papers and not data driven research or investigations.

This article investigates the challenges and achievements of Agile development methods on user experience and user-centered design. I will also use my experience in the software industry to communicate my thoughts from a user experience standpoint and share my experience of transferring from a Waterfall technology to the Agile method

2. BACKGROUND

The case that I have monitored closely for this paper is a midsize enterprise software application very focused on ease of use and despite its modest size, committed to building a strong user experience program. In this endeavor the user-centered methodology was fairly and reasonably practiced in the Waterfall development cycle. The improvements in ease of use in each release were extensively acknowledged by customers and professional press and analysts.

To respond to the fast growing and evolving market, this company's management decided to switch to the Agile method. Following this decision, consultants were hired to help the implementation of this method and training and tutoring staff and workforces. After an initial phase of informational meetings, training "scrums" were formed and product backlogs with stories were created.

Despite the complexity of this process from a development point of view, one of the problems to deal with was how the user-centered design would be implemented through the scrum process and how the limited resources of the user experience team would be allocated to each team.

In the upcoming section of this article, I will review all issues encountered, the way these issues were handled, and a summary of the lessons learned.

3. USER-CENTERED DESIGN IN THE AGILE METHOD

It is fair to say that in the battle to include the user-centered design approach in software development, it took quite a while to just have partial success in convincing consideration of user experience in designing the product. Despite the fact that this approach is not necessarily implemented in all software developments, there is some awareness among professionals about the issue and a growing number of advance technology firms seem to be acutely implementing the process.

When applying the Agile method, it appeared to me that incorporating user-centered design seems even more complicated and not necessarily easier. While participating in this process I have noticed that the literature and communications in this area is still very limited and anecdotal. (Panel discussion, 2006, Mayhew, Constantine, and Lockwood, 1999, 2002, Sy D. 2007, Ambler 2007, Gross et All. 2008)

An example to this would be the books and documentation used to train, inform, and plan Agile mythologies (Schwaber 2004, Poppendieck 2003). In most of these books the user-centered design and its importance in the design process is not adequately explained or covered.

In the case I was involved with, I realized that not only did the books used not cover the user experience aspect, but also the consultants, tutoring teams, and training engineers did not have a good understanding of user-centered design. To some extent their perception of UE professionals was that of prototypers. Consequently the books and literature offered to support training and preparation did not elaborate on how to include user-centered approach into the process or suggest any solutions. There were several questions that needed to be answered:

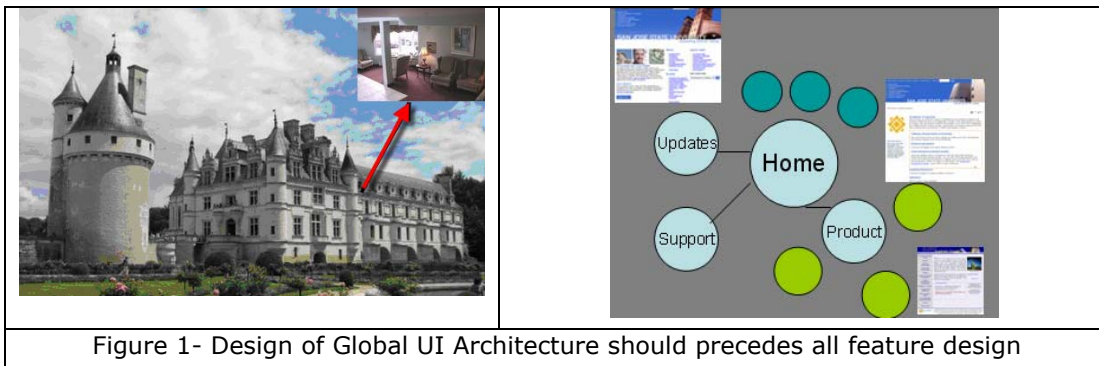
- How and when is user feedback and requirement collected?
- When should user studies be conducted?
- At what stage and how should the general architecture be decided?
- When and how is the detailed design created?
- When and how deep should usability evaluation be conducted?
- How would the limited resources of the UE team be used to cover all needs?

3.1. REQUIREMENT GATHERING AND USER RESEARCH

In most cases, requirement gathering and user studies, even in the smallest projects, require some time. It is unrealistic to define that in an Agile environment all phases of design can be implemented in one sprint or a three to four week cycle. Some experience suggests planning user studies in a separate cycle (SY D. 2007). In my experience, a more in-depth study was very difficult to conduct due to a lack of time and resources. The only possibility was a quick on-line survey by company employees. Participation in user requirement gathering happens to be very difficult.

3.2. GLOBAL ARCHITECTURE VERSUS FEATURE DESIGN

In the Agile methodology it is easy to understand the notion of minimum critical specification or smallest set of requirements and how features are going to be incrementally added to an existing product, based on the customer needs. However, it is harder to define how and when the general architecture of the product should be redefined based on usability criteria. This applies both in redesigning an existing product and designing a new product. Doesn't precipitation of engineering or product management to put a product on the market fast result in designing a product with an overall architecture that is not easy to understand for users? Shouldn't the design focus first on adding a feature even though that might result in building a "Winchester Mystery House"¹? If the answer to those questions is "No" then the design and validation of the overall structure should take precedent to all development processes. One should not decide on the appliances for a kitchen before even deciding where and how big the kitchen will be in a new house construction. In many instances in this process, poor judgment was the result of this type of reasoning. Through this experience I am convinced that a reliable overall architectural design decision should precede all incremental feature designs. It would make sense first to decide what we are building, a "chateau" or a "single family house", before talking about the furniture in the dining room - or in this case the type or property of a widget. Having a drag and drop capability in a page might be considered a "cool" feature by some, but incorporating a new feature in a global, rather than intuitive architecture does not reduce the frustration of the users when they are unable to perform a basic daily task (Fig 1).



¹ The Winchester Mystery House, located in San Jose, California is a mansion that was under construction continuously for 38 years. It was the personal residence of Sarah Winchester, the widow of gun magnate William W. Winchester. Sarah Winchester's wanted construction to continue and never stop. The mansion is famous for its size and lack of a master building plan. It is believed that after the untimely deaths of her baby daughter and husband, Sarah Winchester was convinced by a medium that continuous building would appease the evil spirits of those killed by the famous "Gun that Won the West" and help her attain eternal life.

3.3. DETAILED DESIGN

To provide sufficient time for a more reliable design it appeared to me that it is not a good idea to plan design and implementation as part of one work cycle (sprint). Consequently, I have divided all the work that was involved in design into three phases:

- Conceptual Design
- Interactive Prototyping and Evaluation
- UI Specification and Implementation

Similar experience at Autodesk suggests three cycles: gather customer data, design, and test (SY D. 2007). It would be more efficient to have each be a part of one sprint in order to have enough time to conduct a low fidelity user evaluation at the conceptual level than a more high fidelity prototyping and usability evaluation around the second phase when the implementation has not started yet. In this case we incorporate the user evaluation as a part for each page.

3.4. USER AND USABILITY EVALUATIONS

Reaction time to the problems in Agile is faster. It creates an iteration of the feature but consequently it does not leave enough time to thoroughly evaluate the design with user evaluation. Usability testing requires time to plan, prepare, and conduct. In a short design life cycle it is hard to conduct user evaluations for all features independently and out of context. The user evaluation is more feasible with more general and global features as they provide extensive value. In my experience, the usability evaluation in the small company that I observed was limited to quick informal testing mainly using internal employees due to their availability. Even though one may say having a quick usability evaluation is better than not having any, a quick evaluation does not necessarily help capture all the problems or issues. Constraints blocking implementation reduce the ability to include user input into the design. However, even with quick user evaluations, we were able to identify some major design issues early enough to prevent them from going to development.

3.5. ALLOCATION OF HUMAN RESOURCES

In larger organizations, one would assume that each scrum team can afford to have a dedicated user experience professional thus having resource allocation be less a dilemma of issue versus consistency among the UE designer and infrastructure sharing. In the smaller organization, sharing UE professional time among the different teams might become quite problematic. On one hand, participation in each scrum team meeting becomes very time consuming and reduces the UE productivity time to investigate design or understand users' needs. On the other hand, not participating in a scrum team meeting is somewhat against the Agile principles that require participation of all team members.

In my experience, I came to the conclusion that the user experience team worked as a separate entity serving the development teams. Each scrum team was given a UE professional owner to evaluate their demands and allocate resources by priority. The

UE professional systematically attended the pre-planning and planning meetings to evaluate the work load for each scrum and accumulate a count of all tasks or the work load for the UE team members (visual design prototyping and so on). This helps evaluate the need for outside resources or allocate resources where the value added to the product would be significant.

4. ISSUES AND LESSONS LEARNED

4.1. STORIES - ROLE CONFUSIONS

Even though the scrum team meetings provide an excellent opportunity to discuss issues among team members, understand the problems of each, or learn from the others, they also create a number of confusions in the small corporate environment including: confusion of the roles of scrum master and product owner, unclear and succinct story writing, dynamics of the teams based on specific team member personality, unclear and informal hierarchical relationships, and an illusion of power for some. These were among the problems that I observed in the implementation of the Agile methodology. Even though many advocates of the Agile methodology underline the advantages, one should be aware that in a small company it is important to be prepared to deal with these problems and find the appropriate solutions. Among other things, the most difficult thing that I observed was the lack of a good story. I have also noticed that sometimes stories were written after the task was completed, or the stories were too general or too broad to help a realistic design. (Cohen 2004)

4.2. SELF EVALUATION

Agile offers the team the opportunity to evaluate their actions in retrospective meetings after each sprint, but besides some basic, very practical issues - such as that we should have started this task earlier or later - the major problems and organizational issues were not brought up to the team. Thus it is not realistic to imagine that the people in retrospective meetings will express their wills and criticize the team or organization of the team unless there is a global organizational process to encourage them to address those issues. In some cases I have noticed that the real informal hierarchies created among the team members transformed the team from an active entity to a passive group that tried to synchronize the work load on a daily basis. One team member said one day, "I am just trying to do my job and not have any trouble". This unhealthy hierarchical relationship might not necessarily help to resolve team functioning. This kind of situation becomes really problematic when a company has only two or more products and five or six scrums. The role of product owners running the scrums becomes very unclear and hierarchical.

4.3. SPEED VERSUS QUALITY

Finding an acceptable threshold between quality and speed seems to me to be a major issue in the software development. There is a tendency to let the feature go to the market even if it is not easy to use; "we will take care of it in the next release" is proven to be one of the major issues. It seems to me that this problem is still persisting in the Agile methodology. The only difference is that the next sprint might never come and building a labyrinth of the feature is nearly impossible unless user experience professionals in the teams design the product in a proactive way by

planning the design work ahead of time and thoroughly evaluate it before allocating it to a sprint for implementation. The design should take into consideration incremental feature approaches and take into consideration the way that the product will be released to avoid confusion for the developers and the users

4.4. GOOD AND BAD SIDES

Finding an acceptable threshold between quality and speed seems to me to be an important goal. Those who only see the positive aspects of Agile might end up not seeing the issues and finding solutions for them.

5. CONCLUSION

Agile development can bring users new features incrementally without waiting a long time to get another big release. However, if the user-centered design is not implemented as part of the product design, a more proactive design cycle is not used to create the overall architecture, conduct user studies, and user evaluations, and concepts are not followed, there will be consequences that will illustrate poor usability. Just having Agile development does not necessarily improve usability. A user-centered design, adapted to the needs of the users, is the key to success and better user experience.

REFERENCES

- Agile Alliance, Retrieved Feb 25, 2008, Retrieved from the World Wide Web:
<http://www.agilealliance.com/>
- Agile Programmer Community Blogs, Retrieved Feb 25, 2008, from the World Wide Web:
<http://agileprogrammer.com/>
- Ambler S. W. (2007), Introduction to Agile Usability: User Experience Activities on Agile Development Projects, (2008) Retrieved Feb 25, 2008, from the World Wide Web
<http://www.agilemodeling.com/essays/agileUsability.htm>
- Anderson D. & Schragenheim E. (2003), Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results , Publisher: Pearson
- Boehm, B. , 2002, Get ready for agile methods, with care, Computer, Jan Volume: 35, Issue: 1, Pages 64-69
- Constantine, Larry L. and Lockwood, Lucy A. D (2003), Usage-Centered Software Engineering: An Agile Approach to Integrating Users, User Interfaces, and Usability into Software Engineering Practice, Retrieved from the World Wide Web
<http://portal.acm.org/citation.cfm?id=776931>
- Constantine, Larry L. and Lockwood, Lucy A. D. (1999): Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. ACM Press
- Desilets A. (2005), Are Agile and Usability Methodologies Compatible? HOT Topics! Retrieved Feb 25, 2008, from the World Wide Web
- Gode R. (2006), Usability Engineering with Agile Software Development, Sept. 2006, HCI Vistas Vol-II, Retrieved Feb 25, 2008, from the World Wide Web
<http://www.hceye.org/UsabilityInsights/?p=44>
- Gross J. B., Daughtry III J. M., and Chong Lee J. (2008) Heuristics of the World

Unite! Merging Agile Methods in Software and Interaction Design, 09 February 2008, Retrieved Feb 25, 2008, from the the World Wide Web:

<http://www.agilejournal.com/articles/articles/heurists-of-the-world-unite!-merging-agile->

Hamill P. (2004), Agile User Interface Development, O'reilly on Java.com. Visited on September 2007, www.onjava.com

Manifesto for Agile Software Development, (2001) Retrieved from the World Wide Web:<http://agilemanifesto.org/>

Martin Fowler, The New Methodology, Retrieved Sept 22, 2008, from the World Wide Web:

<http://www.martinfowler.com/articles/newMethodology.html>

Mayhew, Deborah J. (1999): The Usability Engineering Lifecycle: A Practitioner's Guide to User Interface Design. Morgan Kaufmann Publishers

Nerur S. & Balijepally V. (2007), Theoretical and Reflections on Agile Development Methodology, Communication on the ACM, March 2007 Vol. 50, N0

Cohn M. (2004), User Stories Applied: For Agile Software Development, Addison Wesley Professional

Panel discussion (2006), Agile Development Opportunity or Fad?, Chi 2006, Montreal Quebec Canada

Poppendieck M. & Poppendieck T. (2003), Lean Software Development: An Agile Toolkit, Addison Wesley

Schwaber K. (2004), Agile Project Management with Scrum, Microsoft Press

Proceedings of the Agile Development Conference (2005), Pages: 212 – 213

Sy D. (2007), Adapting Usability Investigations for Agile User-Centered Design
Journal of Usability Studies, Volume 2, Issue 3, May 2007, pp. 112-132

Viklund B. (1980), Industrial relations at Volvo

Economic and Industrial Democracy.1980; 1: 125-127