User-Centered-Design in Agile RE through an On-Site User Experience Consultant

Edna Kropp

akquinet

Bülowstr. 66, 10783 Berlin, Germany

Freie Universität Berlin, Institut für Informatik, AG Software Engineering

Takustr. 9, 14195 Berlin, Germany

edna.kropp@akquinet.de

Kolja Koischwitz

akquinet

Bülowstr. 66, 10783 Berlin, Germany

kolja.koischwitz@akquinet.de

Abstract—**This paper is about experiences with user-centered design (UCD) and agile requirements engineering (RE) in fixed-price projects. Efficiently incorporating usability and UCD into RE remains a challenge. In this paper we describe our experience with the introduction of a team role called 'On-site User Experience Consultant' which supports integrating UCD activities tightly in agile RE. We name UCD activities and how they fit in our industrial projects. Also, we describe how old habits in communication and interaction have to change to establish UCD within the project team and with the clients.**

I. INTRODUCTION

The efficient combination of agile software development projects [6] and usability measures [14] is still an unresolved issue for researchers [12][16] and in industrial context [1][4][13][15]. At akquinet we are doing mostly fixed-price software development with explicit focus on UX. These projects, despite being delivered for a fixed-price, are more or less subject to change in the sense that the features finally implemented are not necessarily those agreed upon in the initial contract. Requirements, their details, and their priorities often change throughout the projects. Typically, our projects are complex business-to-business (B2B) solutions. We introduce a role in the development team called 'On-site User Experience Consultant' (osUX consultant). This role helps us to combine agile requirements engineering (RE) and user-centered design (UCD) in our development projects. UCD allows RE to focus on users' experience (UX), as well as their needs and expectations [14]. The information gathered and promoted by the osUX consultant constantly add to the understanding of requirements and the overall goals. Additionally, the osUX consultant assures that the target user groups and their context of usage are in the center of attention all the time, especially for RE and the implementation.

This paper is about experiences with UCD and agile RE in fixed-price projects. We describe the strategies and UCD activities we use to tightly integrate the osUX consultant in agile RE. Additionally, we share our experience on how habits of clients and the project team can obstruct UCD and have to be changed.

II. THE ROLE OF UCD IN OUR AGILE RE

Project management with UCD and agile RE: In our agile fixed-price projects [8] the initial contract consists of a set of rough descriptions of features with their basic, initially known requirements and a clause permitting the client to ask for changes during the project. Together with the agile principle of releasing early and often, this gives the clients a maximum probability to meet their goals without being obliged to exactly define every detail in advance.

During the course of the project our project managers (PM) communicate intensively with the client. The procedures and rules to do so are redetermined for every project. The initial goal of the PM is to establish the individual, best matching communication and documentation for the given project. The project communication not only has to take into account the client's character - is he or she structured or demanding, but also if the client is already acquainted with UCD and aware of its benefits. The clients who are ususally tech-savvy find it easier to be involved in technical decisions of quantitative type, e.g. good at discussing and negotiating numbers, technical solutions, and estimations. Hence, the introduction and integration of UCD can be a challenge throughout the entire development process.

Feasibility in agile projects is based on good estimations [3]. Estimations are usually done by the PM, an experienced developer, and the osUX consultant. A good estimate accounts for every effort of a feature. In our experience, it includes RE and UCD efforts (up to 15%), implementation, testing and bugfixing (up to 20%), PM and communication (10-20%), and a risk surcharge depending on knowledge and complexity.

The initial estimate of a feature set is typically part of the contract and is based on a rough RE and conceptual work. If efforts for a feature increase unexpectedly, UCD is typically the first to be reduced. This leads to weaknesses in usability, despite the client having paid for UX consulting. To prevent this, we negotiate with the client about lowering requirements, e.g. by splitting them into categories such as 'must have', 'should have', and 'nice to have'. Ideally, requirements are refined iteratively during interviews and workshops with the client and ramp-up meetings with the project team before

the corresponding features are implemented. So the core area of communication and interaction between the PM, the development team, and the client is RE [5].

UCD in agile RE: The osUX consultant brings UCD to the RE process. Principles of UCD are also known as usability engineering [10]. This includes input on usability throughout the whole RE process and thereby throughout the whole development process [11]. Our typical UCD activities can be grouped in three phases:

UCD during initiation:
- Research on users and domain
- Visualization - personas and user journey maps
- Scenarios - nontechnical description of use
- User-centered concept of design and interaction

UCD during implementation:
- Reviews on preliminary results
- Feedback on preliminary results

UCD during follow-up:
- Usability tests (UT) - with potential users
- Reviews of solution
- Suggestions for improvements

The initiation phase comprises all UCD activities which need to be conducted before development can begin. Personas and contextual information help to communicate about overall goals. All UCD activities are applied repeatedly depending on the feature or functionality to be developed in the agile project setting. Meetings in the initiation phase (requirements workshop) and just before implementation (ramp-up) help to transfer knowledge supported by UCD activities. For vital, large, and complex requirements a requirements workshop is conducted. Each step in RE such as requirements workshops can be conducted solely by looking at technical aspects of the solution. Therefore our workshops are moderated by the osUX consultant who helps to keep to the following UCD procedure for each feature: identify user roles (personas), write scenarios, specify requirements, their details, their boundaries and priorities with clients with the focus on potential users. A ramp-up is an informal meeting with developers, the PM and the osUX consultant to start a task that is part of the current sprint and has been assigned to a specific developer. These meetings are not planned ahead. They are initiated by a developer. If he or she finds the task complicated and wants help breaking it down into smaller parts, if there is not enough information about it available in the documentation, or if he or she feels the need to consult about implications on the user interface (UI). We usually sit together spontaneously, gather information, write down missing scenarios, discuss issues, and find a consensus about possible solutions.

During implementation, the osUX consultant is an advocate for the users, when conducting e.g. reviews or giving feedback to the developers on preliminary results. All efforts are still included in estimations for the implementation. The further the projects progress the less the influence and the greater the expenses of UCD activities [2]. Our experiences assert that the developers willingness to make changes is higher during implementation. Once they declare the feature to be finished and ready for final review or UT, there is no way to recommend changes – not even for the sake of good UX.

In the follow-up phase UCD activities can help to check whether requirements are fulfilled and the quality of the solution provides good UX e.g. through UTs and reviews. In this phase only usability issues of the highest priority i.e. show stoppers are fixed by developers, because the release is soon after. In this phase the removal of minor usability issues or UCD recommendations depends on the priorities of the client and the management of the PM. Usually, they are gathered by the osUX consultant and communicated as future improvements to the clients.

III. BREAKING OLD HABITS FOR UCD IN AGILE RE

At our company we have taken some measures to establish UCD in our projects such as listing UCD efforts seperately in our contracts, introducing the osUX consultant in the team, and propagating the need for usability to our customers. Although the clients pay for good usability and the team is willing to accept the osUX consultant role, our experiences show that clients and the project team maintain old habits when it comes to performing UCD.

A. Breaking Old Habits with the Client

Leaving conservative origins behind: The typical client (in our case traditional industries such as mechanical manufacturing) is not prepared to participate in an agile UCD project. Often clients have taken part in more formal and sequential development processes and are shaped by this experience. Personal habits, expectations, and approaches as well as existing organizational procedures and requirements at the client's side obstruct agile development.

Introducing an agile approach with UCD and an osUX consultant requires to build up trust and success stories together with the client. Usually, there is no separate budget planned to redo. Hence, there is only little tolerance for mistakes. It is the PM's task and challenge to clarify why the agile way is the best approach, how it is more targeted, effective, and more transparent and why the change from existing approaches to a new one is worth the effort. In order to get agile and user-centered together with the client, the first goal is to find procedures agreed upon to communicate in the project. In our case the following changes were introduced:

- Communication and procedures in workshop are restructured to UCD
- New artifacts such as personas and user-centered scenarios support communication
- New interaction through (UCD) concepts uploaded on wiki to support RE discussions

Too solution oriented: Usually thinking about solutions rather than problems is a good thing. When it comes to UCD this is not the case. Clients tend to think in solutions rather than problems or requirements. They skip important steps of UCD. In the worst case, efforts double or triple, because the conceptual model and the implementation do not match real requirements. To prevent false assumptions to enter the RE process, solutions presented by clients in RE have to be tested from a user's point of view. The best question to ask is: 'What goals of the user are achieved by this?'

The first thing to learn is to start with getting to know the users and their context. At first, UCD is not easy to accept for most clients. Especially, doing interviews, creating personas and describing scenarios seems very theoretical and far from RE. The clients start to value those as important first steps of an elaborated problem solving strategy when they experience their importance in requirements workshops. Workshops are expensive and time is usually short. The goal is to make requirements more concrete and find a consensus about their boundaries. Without sticking to personas and scenarios a lot of discipline by all participants is necessary to stay focused and not be lost in discussions about some detail or solution. Habits do not change over night. Defined UCD procedures in requirement workshops are helpful to avoid quick solutions that may not have a matching use case or scenario.

In a project we did for a machine manufacturer, the client had no possibility to give us access to users due to high competition in the market. Nevertheless the osUX consultant and the PM insisted to stick with the UCD process by creating and refining personas and scenarios. The approach was new and not plausible for the clients. Though, it proved to be beneficial in the following workshop, when filter options for machine entities had to be chosen. At the beginning of the workshop, the client presented his solution of the form by a self-made screen shot consisting of more than 30 different search attributes each having more than a few different values. This solution was set aside in favor of UCD. After collecting all possible scenarios and use cases for filtering, only 6 of the search attributes remained. Without UCD, 80% of the effort would have been wasted and the UI would have been filled with unnecessary options.

Living with unused potential: When it comes to costs, the change to agile RE and UCD is likewise demanding. With fixed price projects, our company takes most financial risks. When refining requirements through RE cycles, the client may expect to get everything the description in the contract could possibly mean. Hence, it is demanding for the PM to communicate to clients where the limits are. The goal is to identify and to implement as much functionality as possible of high priority, must have requirements. The UCD process with its usability tests exposes the imperfection of the software solution more than traditional approches. This often leads to additional expectations on the client's side.

Building up trust through transparency of efforts and close interaction with the client is the best way to handle expectations. We notify the client when we want to reserve or push up

some budget to e.g. conduct a UT or realize some measures resulting from a UT. Often, newly emerging requirements or improvements to existing features can be postponed to later projects or project phases, keeping the current sprint and milestone stable and allowing to finish in time.

In an example from a real project, which consisted of three phases of building a specific feature set validating each by a UT. The first time a new client was confronted with the results of the UT, all weaknesses in usability found were considered as bugs of the software. The client therefore asked for immediate repair of all findings.

Declaring UT results as bugs implicates fixing them is part of the contract might seem logical to the client, but it is not within budget. It is not always advisable to repair every usability issue that is found in a UT [7]. The time spent to make that many changes might prove to be wasted, when in agile projects such as this, requirements may change in the future. Additionally, changes in layout or interaction can cause new, unknown, and unwanted effects on UX. The osUX consultant has the knowledge about the overall goal of the project and is trained to find measures with maximum effect which are minimally invasive. Measures that also best fit in the remaining time and budget. This has to be made visible and understandable to clients to change their thinking about UT results as bugs.

B. Breaking Old Habits with the PM

If the UCD process is added to a project, especially when there is not much prior knowledge in the team, the PM has a lot to steer during the whole project. A lot of the responsibilites for different parts of the "traditional" development process change with UCD. Before UCD was introduced RE was mainly performed by the PM in close collaboration with the clients, e.g. in regular phone calls. In our case, the osUX consultant therefore has a tough time participating in RE and acquiring all knowledge necessary for UCD. Often, the limited time available to conceptualize and implement features is not used as effectively as possible. The PM is stuck in old habits making decisions about usability. These concepts and the time spent in them reduce the time available for the osUX consultant.

There are situations, in which the osUX consultant's role is ignored and work is taken over by the PM. This is the case, when the osUX consultant is not available, the PM is pressured in time, and/or tasks have to be rescheduled spontaneously. There are also situations, in which the PM has to decide to do without or with limited UCD. As stated in the introduction, one possible reason to omit UCD could be that a feature's implementation is getting out of proportion to its estimation. Other times, UCD activities may not prove immediate benefit or are not as easy to be integrated in the development process.

At one occasion the osUX consultant recommended to rethink the navigational structure and labels, although no budget was planned for it. The osUX consultant prepared and proposed a card sorting. Although the PM understood the need for restructuring the navigation, he did not feel comfortable

to ask the client for authorization. There were no real users available to participate and the PM did not believe that anyone from the client's side could represent real users. In the end, the compromise was to discuss the navigational structure in the next workshop. In comparison to personas and scenarios, card sorting remains a UCD activity not established in this project. At another occasion the PM decided - due to time pressure - to give a developer a new task, drew wireframes for it, and introduced them to the development team not conferring with nor mentioning it to the osUX consultant. This resultet in UI elements which did not comply with existing conceptual patterns and reorganizing the UI was not possible due to limited time and budget.

The osUX consultant is the only team member who has a detailed overview over interaction strategies and conceptual patterns. Hence, the PM should never skip to involve the osUX consultant about issues with requirements concerning the usability. The osUX consultant being a bottleneck could be avoided, though it requires the PM to change his habit of doing all RE by himself and giving the osUX consultant heads up about upcoming tasks and the RE process. As described above, the success of integrating UCD greatly depends on the respect and support the role of the osUX consultant gets from the PM.

C. Breaking Old Habits in the Development Team

Introducing and establishing the new role of an osUX consultant is not the same. First it is necessary to introduce the role by advertising UCD activities to all team members including the PM. This can be achieved by offering consulting services whenever possible, e.g. in meetings. The osUX consultant has to find a way to make UCD activities and their results as transparent and accessible to the other team members as possible. The establishment is another challenge. Ideally, the osUX consultant should be available to all team members at all times. Due to different office hours this is not always the case. When the osUX is not available, the team sometimes falls back to old habits and procedures that existed before the osUX was introduced.

It happened, for example, that developers asked the PM for advise when they were not sure about conceptual decisions, because the osUX consultant was out of office. As a consequence, the PM made conceptual decisions that did not match existing conceptual patterns.

It is advisable for the osUX consultant to sit in close proximity to the developers and to regularly ask for updates on current implementation. Then, the osUX consultant has a better chance to realize when old habits are picked up again or steps in UCD are skipped. Also, the developers and the osUX consultant have a better overview of what the other is doing.

The need for a ramp-up shows that although requirements have been formulated and scenarios for features have been written, there are still enough gaps for the osUX consultant to fill before a developer knows exactly what is asked for. Especially in the implementation phase, the osUX consultant keeps the focus on UX and can give valuable advise to the developers. When developers try to solve technical problems their focus is solely on technical issues. In doing so, they sometimes get a tunnel vision and forget about the users and their context [9]. Then the osUX consultant has to adapt the developers' habits to spend all their time with elaborate technical refinement and consider UCD recommendations instead. In these cases, personas and scenarios help the osUX consultant to bring the users back into the developers focus. Placing UCD activities in this manner is less effort than proving obvious weaknesses of UX in a UT and waiting for improvements to be authorized by the client afterwards.

IV. CONCLUSION

In this paper we have shed some light on agile RE with UCD in fixed price projects in industrial context. We named UCD activities we use to integrate UCD into agile RE, e.g. the usage of personas and scenarios in requirements workshops and ramp-ups. Also, we point out how a role called 'On-site User Experience Consultant' is introduced and established in the development team. Finally, we demonstrate in what manner old habits of clients and the project team remain a challenge and how we have coped with those when possible.

REFERENCES

[1] Anitha, P., Prabhu, B.: Integrating Requirements Engineering and User Experience Design in Product Life Cycle Management, In: First International Workshop on Usability and Accessibility Focused Requirements Engineering - UsARE, Zurich (2012)
[2] Bias, R., Mayhew D., Cost-justifying Usability: An Update for an Internet Age, Morgan Kaufmann (2005)
[3] Cohn, M., Agile Estimating and Planning, Prentice Hall PTR (2005)
[4] Ekelund, J., Levingston, C.: Usability in Agile Development, UX Research, Australia (2008)
[5] Ekelund, J., Lowe, D.: Using Partial Designs to Elicit Requirements in Web Development a Survey of Commercial Practice, University of Technology, Sydney (2002)
[6] Fowler, M., The New Methodology, http://martinfowler.com/articles/newMethodology.html (2003)
[7] Krug, S.: Rocket Surgery Made Easy: The Do-it-yourself Guide to Finding and Fixing Usability Problems, New Riders (2009)
[8] Kubitz, T., Agile Festpreisprojekte Risiko oder Chance? (in German), akquinet AG, blog post, http://blog-de.akquinet.de/2014/04/29/agile-festpreisprojekte-risiko-oder-chance-2 (2014)
[9] Lai-Chong Law, E., Hvannberg, E., Cockton G., Jeffries, R.: Introduction to Agile Usability: User Experience (UX) Activities on Agile Development Projects, in Maturing Usability Quality in Software, Interaction, and Value, Springer (2007)
[10] Mayhew, D. J.: The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design, Morgan Kaufmann (1999)
[11] Nielson, J.: Usability Engineering Lifecycle, IEEE Computer vol. 25, issue 3, pp. 12-22 (1992)
[12] Parsons, D., Lal R., Ryu, H., Lange, M.: Software Development Methodologies, Agile Development and Usability Engineering, 18th Australasian Conference on Information Systems, Toowoomba (2007)
[13] Raison, C., Schmidt, S.: Keeping User Centred Design (UCD) Alive and Well in Your Organisation: Taking an Agile Approach, Second International Conference on Design, User Experience, and Usability, Design Philosophy, Methods, and Tools, pp. 573-582, Las Vegas (2013)
[14] Rogers, Y., Sharp, H., Preece, J.: Interaction Design: Beyond Human - Computer Interaction, John Wiley & Sons, West Sussex (2011)
[15] Singer, N.: Intel's Sharp-Eyed Social Scientist (or The Watchful Lab of Dr. Bell), pp. BU1, New York Times (February 16, 2014)
[16] Sohaib, O., Khan, K.: Integrating Usability Engineering and Agile Software Development: A Literature Review, International Conference on Computer Design and Applications (ICCDA), pp. V2-32 -38, Qinhuangdao (2010)