Experiences with User-Centered Design and Agile Requirements Engineering in Fixed-Price Projects

Edna Kropp, Kolja Koischwitz

Experiences with User-centered Design and Agile Requirements Engineering in Fixed-price Projects

Edna Kropp and Kolja Koischwitz

akquinet, Berlin, Germany
Freie Universität Berlin, Institut für Informatik, Berlin, Germany
{edna.kropp,kolja.koischwitz}@akquinet.de
edna.kropp@fu-berlin.de

Abstract. Efficiently incorporating usability and user experience (UX) in agile requirements engineering (RE) remains a challenge. We introduce a team role called 'On-site User Experience Consultant' (osUX consultant) which supports integrating user-centered design (UCD) firmly in agile RE under the constraints of fixed-price software development projects. Our experience shows that such changes in processes cause tensions. These tensions can arise around issues such as the power dynamics within the team, budgets as well as effectively using insights gained from usability tests. We name methods and practices of osUX consulting and how they fit into different phases of an agile RE process. Additionally, we discuss who within a team is ideally suited to occupy this role. Finally, we describe how old habits in the development process have to be adapted to establish this approach within the project team and with the client.

Keywords: User-Centered Design, UCD, User Experience, UX, On-Site UX Consultant, Requirements Engineering, RE, Agile Software Development Process, Dual-Track Process

1 Introduction

The efficient combination of agile software development projects [8] and usability measures [17] is still an unresolved issue for researchers [15][19] and industry [1][6][16][18]. At akquinet we are doing mostly fixed-price software development with explicit focus on UX. These projects, despite of being delivered for a fixed price, are subject to change in the sense that the stories or features implemented in the final solution are not necessarily those agreed upon in the initial contract. Requirements, their details, and their priorities often change throughout the projects. Typically, projects are for business-to-business (B2B) solutions of niche products such as web applications for manufacturers of machines and the domain is complex and not well known. Unlike with business-to-consumer (B2C) solutions we do not interact with our users directly. In our case users are employees or customers of our clients. Therefore, we need alternative ways, such as

user-centered design (UCD), to ensure the users' acceptance and benefits from our solutions.

We introduce a role in the development team called 'On-site User Experience Consultant' (osUX consultant). Typical roles in our project teams are project manager (PM), who is also doing most of requirements engineering (RE), several developers with one dedicated architect role, and a designer (only working part time on the project). The role of an osUX consultant helps us to successfully combine agile RE and a user-centered design (UCD) approach in our development projects. The UCD approach, also known as Human-Centered Design (HCD) [5], allows RE to focus on users' experience (UX), as well as their needs, and expectations [17]. The information gathered and used by the osUX consultant constantly adds to the understanding of requirements and the overall goals. Additionally, the osUX consultant ensures that the target user groups and their usage requirements are at the center of attention at all times, especially for RE and during implementation.

In this paper we describe the strategies and UCD activities we use to firmly integrate the osUX consultant in agile RE. Our experience shows that such changes in the development process cause tensions. These tensions can arise around issues such as the power dynamics within the team, cuts to the UCD budget as well as how to report usability issues while being mindful about budgetary constraints. Moreover, we discuss who within the team, e.g. the PM or one of the developers, is ideally suited to occupy this role.Additionally, we share our experience on how habits of clients and the project team need to be adapted to establish a UCD approach in agile RE.

2   The Role of UCD in Our Agile RE

The basis for integrating UCD in agile RE is to establish a process so that information about the users and evaluation of solutions from a user's point of view become part of the development. In this section we describe how UCD and agile RE are permanently integrated in our fixed-price projects, what our UCD approach contains, e.g. planned methods and best practices, and who should take on the role of osUX consultant.

2.1   Project Management in Fixed-price Projects with UCD and Agile RE

As described in more detail in one of our latest blog posts [10][1], in our agile projects 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. Such changes will be accommodated, provided that the total estimate required to implement changed requirements either does not exceed the estimates before the change or an additional agreement is set up

---

[1] written in German

to cover the additional efforts. Together with the agile principle of releasing early and often, this gives the clients optimal conditions to meet their goals without being obliged to exactly define every detail in advance. This approach allows the project to stay within the predetermined or adjusted budget.

The central topic of communication and interaction between the PM, the development team, and the clients throughout the entire development process revolves around RE [7]. Requirements are typically gathered and documented once and refined iteratively in one or more cycles during planned interviews and workshops with the client. Then, the requirements are further enhanced within the project team (consulting with the client when necessary) in ramp-up meetings. This happens in a sprint before the corresponding features are implemented - comparable to a dual-track process [12][3] - as well as during development, when necessary.

During the course of the project our project managers (PM) communicate intensively with the client. The procedures and rules to do so are set individually for every project and client. The initial goal of the PM is to establish the most suitable communications and documentation for a given project. Communication also depends on how tech-savvy, demanding, and confident the clients are. The integration of UCD in the development process depends on whether the clients have prior knowledge or experience with it.

Feasibility in agile projects is based on solid quotes [4]. These will dictate the day-to-day work of the project team and determine the overall success. Quotes are usually done by the PM, an experienced developer, and the osUX consultant. A good estimate covers all labor needed to develop a feature. From our experience estimates (including an explicit budget for usability measures) comprise of the following:

- The cost of calculating the quote itself
- RE and UCD (up to 15%)
- Implementation
- Testing and bug-fixing (up to 20%)
- PM and communication (10-20%)
- A budget for contingencies

The initial estimate of a feature set is typically part of the contract and is based on an initial rough RE and conceptual work before and during estimation. Its main purpose is to give the project team and the clients a foundation for all upcoming communication.

2.2 UCD Approach - Methods and Practices of an OsUX Consultant

The osUX consultant brings a UCD approach to the agile RE process. This includes input on usability and UX throughout the whole RE process and hence

throughout the whole development process [14]. UCD activities can be grouped for each feature in four phases: initiation, conceptualization, implementation, and follow-up. Below we have listed crucial UCD activities in each phase. There may be other, exceptional phases in our development process such as bug-fixing or release phases which are not included in this list of phases due to their irregular appearance. The further the projects progress the less the influence and the greater the expenses of UCD activities [2].

UCD in Initiation

- Project planning - within project team
  - Setting goals for UCD
  - Estimation on initial requirements from a UCD point of view
- Contextual inquiry - with potential users and other stakeholders
  - Research on users, domain, and context
- Visualization and modeling of users in their context and domain
  - Personas - concrete user representatives
  - Work-flow analysis
  - Scenarios - nontechnical description of use
- UCD moderated requirements workshop - with client, potential users, and other stakeholders

UCD in Conceptualization

- Allocate and analyze all existing information about a planned feature or requirement
- Concept of design and interaction
  - Transform requirements and scenarios into use cases
  - Transform use cases into concepts - scribbles, wire frames, paper prototypes
  - Match concepts with existing style guide and patterns
  - Allocate feedback from client, potential users, and other stakeholders
- Ramp-up meeting - break down complex tasks within team

UCD in Implementation

- Formative evaluation
  - Expert reviews on preliminary results
  - Paper prototype tests
- Feedback about UX to team and client on preliminary results

UCD in Follow-up

- Summative evaluation
  - Expert reviews on existing features
  - Usability tests (UT) - with potential users

- Reporting about UX to team and to clients
  - Summary of the overall usability of the solution
  - Suggestions for future improvements

The **initiation phase** comprises all UCD activities which need to be conducted before conceptualization and development can begin. Visualizations and models of contextual information help to communicate about overall goals and concrete requirements. Some activities in the initiation phase such as initial estimations and contextual inquiry are performed once and most of their results apply throughout the entire development process. Most other UCD activities are applied repeatedly depending on the feature or functionality to be developed in the agile project setting. Meetings in the initiation phase, e.g. requirements workshops, and just before implementation, e.g. ramp-up meetings, help to transfer knowledge between stakeholders supported by the osUX consultant and UCD activities. For vital, large, and complex requirements a requirements workshop is conducted. The workshop is moderated by the osUX consultant who provides a well structured approach and keeps focus by avoiding unnecessary discussions. The structured approach includes identifying user roles through personas, writing scenarios, specifying requirements, their details, their limits and priorities with clients, potential users, and other stakeholders.

Sometimes there are several months between RE workshops and the **conceptualization phase**, hence good documentation during initiation is needed. In preparing of the concept for a planned feature all relevant information, such as information gathered during contextual inquiry about users, given requirements, created scenarios, need to be collected.

The **implementation phase** starts with a ramp-up meeting. An informal meeting with the PM, the osUX consultant, and the developers who plan to start a task that is part of the current sprint. These meetings are not planned ahead. They are initiated by a developer, if the task is complicated or information is missing. Activities in this meeting are information gathering, knowledge exchange, writing down missing scenarios, discussing issues, and finding a consensus about possible solutions. In this phase the osUX consultant is an advocate for the users, when conducting e.g. reviews or giving feedback to the developers on preliminary results. All these efforts are still included in the quotes for implementation. The developers' willingness to make changes is greater during this phase as opposed to when the task is declared completed. Once the developers declare the feature to be finished and ready for a final review or a usability test, there is usually neither the budget nor the willingness for changes due to usability issues or UX recommendations.

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 usability testing and summative evaluation, e.g. expert reviews. In this phase only usability issues of the highest priority i.e. show stoppers are fixed by developers, because the release is close. The removal of minor usability issues or UCD recommendations then 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 client.

There are conditions under which it is especially demanding to properly integrate UCD activities in the project phases described above. During initiation the clients present a rough set of requirements. These are usually feature-driven and do not have a specific focus on users' needs or expectations. This leads to situations in which important UX requirements and risks can easily be overlooked.For standard tasks, e.g. sign-in processes or user administration, overall quotation and initial conceptualization is easy. But for complex tasks and individual solutions, e.g. complex UI elements such as calendars, time bars, or charts, the project team has to invest additional time to analyze the clients' input from a UCD perspective and check for hidden complexity before proceeding with quotation and conceptualization.

During the implementation phase the team follows UCD methods presented above, e.g. initial concepts and ramp-up meetings, as long as they are within the estimated budget. Conflicts arise when there are additional demands from clients, which often emerge with a tangible solution, but not enough time or resources to adequately handle them. This leads to short-cuts in the process which are prone to weaken the already achieved usability of the current solution. Similar effects are observable in the release and the bug fixing phases. With limited resources and deadline pressures, quick fixes are made without usability considerations. Most times these have unforeseen implications on the usability of the current solution.

2.3   Who should take on the role of On-site UX Consultant?

When considering who is suited to take on the role of osUX consultant, it is necessary to look at the skills this person should have. Among many others, these are:

Trained in UCD methods and practices - The osUX consultant is well trained in UCD methods and practices and knows most state of the art UCD procedures. Hence, she or he can make decisions such as what UX evaluation method, e.g. formative or summative, usability test or expert review, etc., should be used in what stage of the process. And she or he can perform them without additional supervision.

Availability - The osUX consultant should be available throughout the entire development process. Hence, she or he knows about every initial request or change in requirements, knows about existing styles and patterns and the reasons they were chosen, always looks out for potential optimization (for better UX and solutions that fit into budget), can choose from best fitting UCD methods and practices/ state of the art UCD procedures, and can support the team with fast solutions. Alternatives to having an osUX consultant role or person available throughout the entire development process are to outsource to an agency or an inhouse service for conceptualization and evaluation, and other variations to those approaches. The advantage of having an osUX consultant is that open

issues and questions are dealt with immediately or at least within the specified deadlines.

Responsibilities to users and self-assigned tasks - The osUX consultant focuses on the quality of the software solution concerning UX. This includes the initiation of the right UCD activities despite any skepticism of other project members and to be an advocate of users' needs and expectations even when the project suffers from pressure in time or budget. Her or his objective is to satisfy the user's needs (not the client's) in the best interest of the client. The responsibilities and assigned tasks of the osUX consultant are limited. She or he can only introduce recommendations and offer her support to the project. Thus, the PM and the client make final decisions on how to use the budget. Also, the developers decide when and if to ask for help and advice. Additionally, all initial plans and quotes, such as conducting a usability test or using 10% of the planned budget for UCD methods, are always at risk of omition through pressures of time and budget.

Introduction of improvements - The osUX consultant has typically the unpleasant role to point out weaknesses of existing requirements or solutions. Also, she or he has to formulate recommendations that have to prove their worthiness to the user. This requires some distance to the development process as well as a lot of diplomatic skills. Sometimes, the client's wishes have to be questioned or at other times solutions have to be challenged by e.g. usability tests.

Besides the personal skills of the osUX consultant, it is important to consider what position, i.e. responsibilities and authority, the person has within the project.

The advantages to assigning the role of an osUX consultant to a team member such as the PM or a designer are substantial. The PM is already a fixed role in any project for the entire duration. Additionally, the PM typically has deep insight into requirements, because she or he is in charge of RE. UCD activities are then more likely to be conducted, because next to the client the PM has the authority to plan the budget and to make final decisions in the project. Assigning the role of osUX consultant to the designer could be favorable as well. It is likely that she or he is at least part-time in the project already and is interested in and acquainted with most of the visual aspects of good usability, e.g. supportive positioning of content or buttons - maybe even some of the interactional aspects, e.g. a supportive navigational structure.

Despite these apparent advantages, our experience shows, however, that in reality, it is preferable not to have a current team member, including developers, in the role of osUX consultant. They each have their own focus and responsibility which generally collides with the responsibilities and assignments of an osUX consultant. It was observable that, even with an osUX consultant on the team, the PM would omit UCD activities, among others, in the face of time or budgetary pressures. Also, in a B2B project it is the PM's first priority to please the client, not to satisfy the users' needs who are e.g. employees of the client. This might be different when looked at in contexts other than fixed-price

projects for B2B solutions, e.g. when revenue is generated directly by a B2C plat-form. Additionally, none of the above mentioned team members are trained in UCD methods or practices, because usually training in other areas, e.g. project management, technology, graphics tools, etc., have higher prioritiy. Also, it was observable and reported by the team members such as architects, designers, and developers that they get a sort of 'tunnel vision' and therefore lack the necessary distance to adopt a usability perspective. When they focus and spend a lot of time with a task, even the most complex concepts seem intuitive. They tend not to take the users' context into account any more.

3   Breaking Old Habits for UCD in Agile RE

One might assumed that little would stand in the way of a solution with good usability and UX, when clients support a UCD approach by committing up to 15% of explicit budget for UCD activities, which are fully support by all project team members and management. However, we observed that independent of best intentions old habits of the client, the PM and other team members such as developers have to be addressed first.

3.1   Breaking Old Habits with the Client

Conservative attitude – The typical client (in our case traditional industries such as mechanical manufacturing) is not prepared to participate in an agile project with a UCD approach. Often clients have taken part in more formal and sequential development processes in which the focus is mainly on budget, time and scope, rather than on users' needs and expectations, and hence they are shaped by their experience. Personal habits, expectations, and approaches as well as existing organizational procedures and requirements at the client's side stand in the way of agile development.

Introducing an agile approach which includes UCD and an osUX consultant requires to build a trusting relationship with the client and create success stories. Usually, there is no budget allocated to start over, hence there is are only small margins for errors. It is the PM's task and challenge to convey why the agile way is the better approach, how it is more effective and transparent and why the change from existing approaches to a new one is worth the effort.

In order to maintain an agile and user-centered approach, it is important to establish rules of engagement with the clients that will ensure a focus on these aspects. The following questions should to be answered:

- Are there regular meetings, what is their structure, what is documented by whom, when, and where?
- Given the project plan, how do we approach the features to implement next?
- How do we handle and how do we refine basic assumptions stated by our contract?
- How do we dissolve different interests and opinions?

- What happens when requirements change?

As a result of a development process with a UCD approach RE workshops can be restructured to be more user centered (described in section 2.2). Additionally, other tools, e.g. personas and scenarios, can dissolve some conflicts in communication. Making features tangible through visualization helps with some communication issues.

Solution oriented – Clients tend to think first and foremost of the end-product, e.g. they present a screen shot instead of stating users' needs and usage requirements. This means that they may skip important steps of the UCD approach. Usually, clients do not share the same interests in the solution as the target user groups and hence miss important factors for a solution with good UX. This increases efforts, because the conceptual model and the implementation do not match real requirements. To prevent false assumptions from entering the RE process, presented solutions have to be tested from a user's point of view. The best questions to ask are:

- What problems of the user are solved by the solution presented by the client?
- What part of the solution presented by the client might be irrelevant to users and could be left out?
- What tasks would the user perform to achieve her or his goals?
- How would the user most likely perform her or his tasks, in what order, can content be semantically grouped?

The first thing to learn is to start with getting to know the users and their context. The UCD approach is at first not easy to accept for most clients. Especially, doing interviews, creating personas and describing scenarios seems very theoretical and far from the problems to be solved. 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 procedures in e.g. requirement workshops are helpful to avoid e.g. quick solutions that may not have a matching use case or scenario.

In a project we conducted 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 client. However, the UCD approach proved to be an immediate benefit in the following RE workshop. The task was to define filter options for machine entities. The client began the RE workshop by presenting his solution, a self-made screen shot, consisting of more than 30 different filter attributes each having multiple possible values. This solution was set aside in favor of the UCD approach. We started by collecting all possible use cases for

filtering. Then, we only allowed filter options which matched the allocated use cases. Afterwards, only 6 of the formerly presented search attributes remained in the final version. This saved 80% of development efforts and enhanced the UI's usability by omitting over 24 additional unnecessary filter options.

Unused potential – Clients need to get accustomed to the fact that some UCD activities such as usability tests (UT) or recommendations from expert reviews uncover potential weaknesses or improvements which cannot all be fixed or implemented within the estimated budget. 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. Of course, we trust our clients not to exploit the offer. In the contract efforts are pinned to roughly described features and when refining them through RE cycles, the client cannot expect to get everything an initial description could have possibly meant. Instead, the estimated budget for a single feature should be seen as a limit. This limit states the capacity to implement as much functionality as possible starting with high priority, must have requirements. Suggestions on how to cope with unused potential are:

- Build up a trustful relationship with your client
- What makes a solution complete? Find mutual agreeable basis for the scope of planned features and emerging improvements.
- Establish definitions to distinguish between issues which are bugs, weaknesses in usability and suggested improvements

With every client we have to build up trust through transparency, constant communication, and close interaction. Transparency is needed on estimated and real efforts, of saved efforts, and overspent time. 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.

We were surprised by the client's reaction in one of our projects. We planned three milestones each ending with a UT. After the first UT, the client declared all usability issues found as bugs and therefore asked for immediate repair of all findings.

Performing all measures would have gone well beyond all estimates in the contract. Declaring all UT results as bugs and hence assuming to fix them is part of the contract seemed right to the client. Yet, it is wrong in more than one way. In general, it is not advisable to repair every little weakness that is found in a UT. 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. Moreover, even the smallest changes in layout or interaction can cause new, unknown, and unwanted effects on UX.

The osUX consultant has the knowledge about the overall goals 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.

3.2 Breaking Old Habits with the PM

When a UCD approach is added to a project, especially when there is not much prior knowledge about UCD in the team, the PM has a lot to steer during the whole project. A lot of the responsibilities for different parts of the 'traditional' agile development process change with UCD. Before UCD was introduced, RE was mainly performed by the PM in close collaboration with the client, i.e. in regular phone calls without a lot of documentation. The osUX consultant therefore has a tough time participating in RE and acquiring all knowledge necessary for a UCD approach. Often, the limited time available to conceptualize and implement features is not used as effectively and efficiently as could be. The PM is held back by old habits making decisions about usability although this is the task of the osUX consultant. These concepts and the time spent in them reduce the estimated time available for the osUX consultant. Issues to be addressed by the PM, when an osUX consultant becomes a new team member:

- Which parts of the process need to be adjusted to UCD and the osUX consultant?
- What responsibilities and tasks are now assigned to the osUX consultant which were originally performed by other team members?
- Who decides on what UCD activities are planned and performed? Who has the knowledge and the authorization to make those decisions?
- Who should participate in UCD activities? Only the osUX consultant, the client, the project team, or all stakeholders?

The process comprising UCD needs to be adjusted to a dual-track process in which there is an explicit conceptual phase before implementation of important and complex tasks can be started. Also, the osUX consultant has to be integrated into existing communication channels concerning RE. Additionally, the PM needs to refer discussions about usability and UX to 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 section 2.1, one possible reason to omit UCD activities could be that a feature's implementation is getting out of proportion to its estimation. Other times, UCD activities may not provide immediate benefit or are not as easyly integrated in the development process.

Not all UCD activities can prove immediate benefit and are as easy to be integrated in the development process as e.g. use cases. At one occasion in a project with an alpha-state solution, the osUX consultant recommended to re-think 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, a compromise was to discuss the navigational

structure in the next RE 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 and the absence of the osUX consultant - 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 resulted in UI elements which did not comply with existing conceptual patterns. Reorganizing the UI to be consistent to existing patterns was not possible due to limited time and budget.

The osUX consultant is the only team member who has expert knowledge about usability and UCD as well as a complete 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 support the role of the osUX consultant gets from the PM.

The PM learned about particular aspects of usability considerations as far as they concerned issues that arose during the project. However, the issues to solve were specific to the project and different every time. Therefore, it cannot be expected of him to come up with usability considerations apart from the ones he witnessed. More importantly, during the course of the project, the PM understood the UCD approach and its purpose better, got to know some of its benefits and experienced the consequences of omitting it.

3.3   Adapting 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 long term establishment of the osUX consultant role 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. UCD activities can become a bottle neck when the osUX consultant works part-time or other team members work overtime. When the osUX consultant is not available, the team sometimes falls back to old habits and procedures that existed before the osUX consultant was introduced in the team. The following methods can be used to establish the role of an osUX consultant in the development team. The osUX consultant should:

- show positive outcomes of UCD activities, e.g. solutions that are based on existing patterns or general usability considerations that will save developers time

- make usability considerations as transparent as possible, so developers can learn and apply them without feeling patronized
- consider to integrate the whole team in conceptual decisions. So it is less likely that whom ever had a similar role (intrinsic or officially assigned) before will have reason to work against the new approach.

It is advisable for the osUX consultant to sit in close proximity to the developers. 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 advice 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 [11]. Then the osUX consultant has to adapt the developers' habits to spend all their time with elaborate technical refinement and consider basic 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.

Here, the same examples as mentioned in section 3.2 apply as good examples. The developers asked the PM for advice because the osUX consultant was out of office, when they were not sure about conceptual decisions and the PM made decisions that did not match existing conceptual patterns. Also, the developers accepted wireframes drawn by the PM not complying with existing patterns due to time pressure. Additionally, more than once developers build small add-ons to existing views that were demanded by the client, once the client saw existing solutions. These solutions did not comply with existing patterns such as e.g. static information that was placed at positions reserved for controls, and styled in colors, e.g. in bright yellow, that were not defined in the style guide.

4   Conclusion

In this paper we shed some light on our experiences with agile RE in fixed-price projects in the B2B industry. We point out how a new role called 'On-site User Experience Consultant' (osUX consultant) is introduced and established in the development team. We suggest the role of an osUX consultant to be taken by a separate team member who is trained in UCD methods and practices and who focuses on users' needs and expectations (with respect to clients' wishes) and is available for usability consulting and feedback throughout the entire development process.

Also, we name UCD activities we use to integrate a UCD approach into agile RE, e.g. the usage of personas and scenarios in requirements workshops and

the establishment of ramp-up meetings before implementation to break down complex tasks together with developers, the PM, and the osUX consultant. Recommendations made after implementation usually end up as improvements for later milestones. It can be especially demanding to properly integrate UCD activities in some phases of the development process. For example, during the initiation phase estimation and initial conceptualization for complex or individualized solutions are prone to underestimation and hidden complexity. Also, if the clients have additional demands, which often emerge with a tangible solution and which are not included in the estimated budget, this can lead to a lack of usability due to short cuts during the implementation, release, and bug-fixing phases.

Finally, we have demonstrated in what manner old habits of clients, the PM, and the development team remain a challenge and how we have coped with those so far. Although clients accept an agile development process, they are still conservative in project communication and decision making. Also, clients tend to be solution-oriented, i.e. presenting solutions in form of e.g. self-made screen shots instead of describing what problems to be solved. Additionally, clients need to learn to live with unused potential, e.g. results of a usability test should not be treated as bugs in general. Results should be seen as the potential to further improve usability. Before UCD was introduced the PM used to do all the RE by himself. Hence, communication was directly between PM and clients and documentation reduced to a minimum. The PM decided when to forward what information to team members. With UCD the PM needs to change communication habits and enlarge documentation to ensure earliest possible knowledge exchange with the osUX consultant. We learned that introducing and establishing the role of an osUX consultant in the development team are two separate steps. After introduction the osUX needs to advertise UCD services, e.g. usability consulting on specific tasks, to each team member and establish new communication channels for evaluation and feedback on usability, e.g. in team meetings.

In the future, we hope to further explore the existence of generalizable phenomena, i.e. habits to change. It is also planned to incorporate how these phenomena manifest themselves in the process and what their consequences are. Where possible, we will additionally support our findings with a quantitative approach, e.g. how much effort it took or saved to overcome old habits.

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. Cagan, M.: Dual-Track Scrum, http://www.svpg.com/dual-track-scrum/ (2012), last viewed March 12th 2015

4. Cohn, M.: Agile Estimating and Planning, Prentice Hall PTR (2005)
5. DIN EN ISO 9241-210: Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems, http://www.iso.org/iso/home/ (2010), last viewed March 12th 2015
6. Ekelund, J., Levingston, C.: Usability in Agile Development, UX Research, Australia (2008)
7. Ekelund, J., Lowe, D.: Using Partial Designs to Elicit Requirements in Web Development a Survey of Commercial Practice, University of Technology, Sydney (2002)
8. Fowler, M.: The New Methodology, http://martinfowler.com/articles/newMethodology.html (2003), last viewed March 12th 2015
9. Krug, S.: Rocket Surgery Made Easy: The Do-it-yourself Guide to Finding and Fixing Usability Problems, New Riders (2009)
10. Kubitz, T.: Agile Festpreisprojekte Risiko oder Chance?, akquinet AG, blog post, http://blog-de.akquinet.de/2014/04/29/agile-festpreisprojekte-risiko-oder-chance-2 (2014)
11. 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)
12. Lent, J.: Dual-track Agile keeps requirements in check, http://searchsoftwarequality.techtarget.com/opinion/Dual-track-Agile-keeps-requirements-in-check (2014), last viewed March 12th 2015
13. Mayhew, D. J.: The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design, Morgan Kaufmann (1999)
14. Nielson, J.: Usability Engineering Lifecycle, IEEE Computer vol. 25, issue 3, pp. 12-22 (1992)
15. Parsons, D., Lal R., Ryu, H., Lange, M.: Software Development Methodologies, Agile Development and Usability Engineering, 18th Australasian Conference on Information Systems, Toowoomba (2007)
16. 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)
17. Rogers, Y., Sharp, H., Preece, J.: Interaction Design: Beyond Human - Computer Interaction, John Wiley & Sons, West Sussex (2011)
18. Singer, N.: Intel's Sharp-Eyed Social Scientist (or The Watchful Lab of Dr. Bell), pp. BU1, New York Times (February 16, 2014)
19. 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)