

Agile Usability Patterns for UCD Early Stages

Ana Paula O. Bertholdo¹, Tiago Silva da Silva², Claudia de O. Melo^{1,3},
Fabio Kon¹, and Milene Selbach Silveira⁴

¹ IME/USP - Universidade de São Paulo

{ana,fabio.kon}@ime.usp.br

² ICT/UNIFESP - Universidade Federal de São Paulo

silvadasilva@unifesp.br

³ Thoughtworks

cmelo@thoughtworks.com

⁴ FACIN/PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul

milene.silveira@pucrs.br

Abstract. The integration between agile methods and UCD has been addressed by several authors in recent years. However, a gap remains regarding how the practices have been described, lacking a standard that both designers and agile practitioners can understand and apply.

This study aims to propose agile usability patterns based on the literature, with a focus on the User-Centered Design early stages. The goal of the proposed patterns is to facilitate the use of the best agile usability practices by identifying more clearly in which context the pattern can be applied, and what is the problem that each pattern solves, presenting examples.

Keywords: agile usability, agile UCD, agile UX, best practices, patterns.

1 Introduction

The adoption of agile development methods has grown steadily in the software industry, since their creation [1, 2]. According to Larman [3], agile methods apply time boxed iterative and evolutionary development, adaptive planning, promote evolutionary delivery, and include other values and practices that encourage agility – rapid and flexible response to change.

Agile development values are in accordance to the underlying principles of User-Centered Design (UCD) [4], e.g., focusing on individuals and interactions and collaborating with customer. By integrating UCD and agile methods, we can ensure the software produced adds value to the business and to the end user. As observed by Ratcliffe and McNeill [2], there is no motivation in using software if it does not add value.

With regard to usable practices or methods, agile methods are considered lightweight, *i.e.*, the team should only do what is necessary to bring value to the customer. On the other hand, UCD shows the importance of real users in system development. If used together, UCD and agile methods will benefit, since

software is developed by and for people. This integration between agile methods and UCD has been addressed by several authors in recent years. For instance, Sy [5], Fox *et al.* [6] and Silva *et al.* [7] arrived at very similar proposals.

However, it is still necessary a breakdown of each stage of these proposals. Moreover, a gap remains regarding how the practices have been described, lacking a standard that both designers and agile practitioners can understand and apply. Patterns can help in understanding how to integrate the practices of the two communities involved and allow us to visualize the commonalities shared.

This study aims to propose agile usability patterns based on the literature [8], with a focus on the User-Centered Design early stages: Identify need for human centered design; Specify context of use; and Specify requirements [4]. The research question for this study is: what are the best usability practices used in the agile methods community related to UCD early stages?

The remainder of this paper is structured as follows: Section 2 presents key concepts for the research and a brief summary of recent related work; Section 3 presents how the synthesis and collection of data was performed to rate usability practices in patterns; Section 4 presents the Agile Usability Patterns related to UCD early stages; and Section 5 brings up conclusions as well as future directions for the research.

2 Background

Agile methods entail adaptive planning and iterative processes. Having an iterative process means a project is divided into subsets of features, unlike the waterfall that divides a project based on requirements analysis, design, coding, and tests. In the iterative style, the development cycle is composed of a series of small incremental releases. Each release is set at regular intervals, typically from two to four weeks, which are called sprints in Scrum [9] or iterations in eXtreme Programming (XP) [10], in which the team carries out a complete development cycle of a subset of the requirements.

There is an international standard that is the basis for many UCD methodologies. This standard (ISO 13407: Human-Centred Design Process) defines a general process for including human-centered activities through a development life-cycle, but does not specify exact methods. In this model, once the need to use a human centered design process has been identified, four activities form the main cycle of work: (i) Specify Context of Use; (ii) Specify Requirements; (iii) Produce Design Solutions; (iv) Evaluate Designs.

While usability methods have focused primarily on the typical users of the system that actually use it, agile methods are concerned with the fulfillment of customer needs, more related to the system business values. The concern for the users therefore is present in both, however they differ with respect to the type of user considered. Thus, the integration of these areas benefited the users, whatever the type or profile they belong.

The systematic literature review on UCD and agile methods conducted by Silva *et al.* [7] presented 58 studies addressing this topic. Some of them depict

an overall picture for the integration of UCD and agile methods, such as Fox *et al.* [6], and Sy [5]. 31 studies of the systematic review addressed the early stages of the UCD. However, classifying the studies according to the UCD stages is a hard task. Due to the agile projects iterative nature some UCD stages overlap.

The literature suggests that for a smooth integration between UCD and Agile, UCD should take advantage of the iterative nature of Agile. For instance, Sy [5] suggests that breaking designs down into cycle-sized chunks gives them the freedom to mix and match different types of usability investigations into the same session, which enables them to juggle more than one design, and more than one type of usability investigation at the same time. In order to do so, UCD should be performed aiming at applying all activities of the UCD cycle for each subset of features. This is the aim of Agile UCD, breaking UCD stages down into the agile cycles size.

3 Method

The goal of data collection was to find the usability practices most used by the agile community. The search criteria was defined as follows:

- ((“usability” OR “usability methods” OR “User Centered Design” OR “User eXperience” OR “Human-Computer Interaction” OR “Computer-Human Interaction”) AND (“agile methods” OR “agile development” OR “eXtreme Programming” OR “Scrum” OR “agile”)).

The filtering process consisted of: (*i*) Reading the title, (*ii*) Reading the summary, and (*iii*) Reading the complete study. For each phase, the studies that were not in accordance to the inclusion criteria were excluded.

The synthesis of the collected data was performed by checking the items that define a pattern – name, context, problem, solution and examples – in each practice. Practices that did not have the data for each of these items were excluded from the pattern. A pattern would need to have at least three similar practices for the same problem. Furthermore, for a practice to be defined as a pattern example, it should belong to different projects among the pattern examples considered.

4 Agile Usability Patterns for UCD Early Stages

In this study, a pattern is defined according to Alexander *et al.* [11], *i.e.*, a pattern is a structured method of describing good design practices within a field of expertise. Still following Alexander’s definition [11], we describe the patterns by presenting (*i*) Name; (*ii*) Context; (*iii*) Problem; (*iv*) Solution; and (*v*) Example.

The patterns are presented according to the UCD stages [4], which facilitates the understanding of the goal of each pattern described. This paper focuses only on the usability practices which comprises the following steps of the UCD: (1) Identify Needs for Human-Centered Design; (2) Specify Context of Use; and (3) Specify Requirements.

Tullis and Albert reported the difference between the terms usability and User Experience (UX): "Usability is generally considered the user's ability in using something to accomplish a task successfully, while user experience has a broader view, looking at the full interaction of the individual, as well as their thoughts, feelings and perceptions that result from this interaction" [12].

In the agile community, it is extremely common to use the term User Experience to any activity related to user research, UCD, usability, or interface design. Therefore, in the patterns described below, we use the same term. However, the term not necessarily indicates that all UX activities were really performed.

Based on the method defined in Section 3, we propose the following set of Agile Usability Patterns related to UCD early stages.

4.1 Identify Needs for Human-Centered Design

Pattern: *Sprint Zero*

Context: Sprint Zero is a practice that aims at better defining a broader view of the project. It is a stage before starting the project implementation, to define product view and general goals, to roughly plan future sprints, to define design principles and to clarify roles of the team members and communication methods.

It is also a critical sprint to do any work of UX or usability before the start of code production. In the agile community, it is usually performed by a UX or usability team, composed of usability specialists, designers, and UX managers who study the user and draw some screens to communicate the raised ideas. Some teams involve other members, such as developers.

Problem: Missing the "big picture" of the system in the beginning of the development with agile methods. In addition, priorities might be unclear and team members might work with wrong things because they did not understand which the real priorities were. Therefore, they report unnecessary work. Also, there is little time for upfront design. The main forces involved are:

- **Force 1:** The UX team needs to clarify the "big picture" of the system.
- **Force 2:** Short time to understand the "big picture" of the system during the Sprints, where there are parallel UX and development activities.

Solution: A short Sprint before the code implementation to define a broader view of the product, general goals, to roughly plan the next sprints and to define design principles.

Example: Chamberlain *et al.* suggest Sprint Zero should be done before the Planning Game, an agile planning meeting, so that usability aspects can be discussed during the Planning Game [13]. Belchev and Baker report contextual inquiry being used as one-to-one interviews conducted in the users workplace, which aim at observing ongoing work [14].

Pattern: *One Sprint Ahead*

Context: The iteration time is considered too short to perform all the UX activities and implement their results. Therefore, UX professionals need to work

one sprint ahead of the development team. Thus, the UX team is able to conduct user research and produce design solutions to the upcoming development iteration. At the same time, the UX team evaluate the already coded functionalities from the previous iteration. In this context, there are two work tracks within the project: UX and development.

Problem: Making the development and the UX team synchronized, then both can collaborate and provide input to the development workflow. Main forces involved:

- **Force 1:** Developers need design definitions and system requirements to analyze and implement functionalities.
- **Force 2:** The UX team needs to evaluate designs implemented by developers according to user needs.
- **Force 3:** The UX team needs to interact with users and customers to define design requirements and solutions.
- **Force 4:** Iteration usually offers short time to perform the activities described in the three first forces.

Solution: The UX team works at least one iteration ahead of development team.

Example: The systematic review conducted on UCD and agile methods [7] described several articles [13, 15, 16] that address the One Sprint Ahead issue. They reported the use of one sprint ahead concept, that in summary consists of UX or usability specialists performing tasks related to user research, design and user interface evaluations while the development team implements the code of stories designed in the previous sprint and fix problems found by the UX team.

Pattern: *UX Specialists as Product Owners*

Context: Customers usually provide business requirements that the system need to attend. When hybrid versions – composed by Scrum and XP – are followed, Product Backlog items are described as User Stories. The adoption of UX specialists as Product Owners means that real users interests will be taken into account as well as customers interests. The Product Owner shares user needs with the team and the team defines what functionalities they will be able to finish until the next Sprint. Software development with agile methods is iterative or divided into iterations or sprints.

Problem: Integrating business requirements into usability requirements. Main forces involved:

- **Force 1:** Adding real users needs to a development environment without leaving business requirements behind.
- **Force 2:** Combining customers and real users needs.

Solution: UX specialists composing a Product Owners team.

Example: Gothelf describes his experience as Product Owner in an agile team using Scrum [17]. He thinks that UX designers work is the link between business

and customer. By addressing this information under the view of the Product Owner, it is possible to come up with ideas for design early and accordingly with business requirements.

Pattern: *Users Time is Valuable*

Context: In agile methods, all the activities are related to a subset of functionalities at each iteration or sprint. Therefore, when carrying out user research, it is necessary to save some time for gathering requirements for the upcoming iterations and evaluations of the former ones. From a single visit, it is possible to use time for several activities with users. In general, the activities related to users are performed in a face-to-face format and only by members of the UX team.

Problem: Short time to conduct usability tests and user research. Main forces involved:

- **Force 1:** UX activities are performed during sprints or iterations.
- **Force 2:** Sprints or iterations are too short to be divided into a specific type of activity (evaluation, design or research) at each meeting with the user.

Solution: Each meeting with typical users is used for several activities, such as usability tests of the design developed in the previous iteration and user research to define requirements for the upcoming iteration.

Example: Patton states that the practice called “Leverage user time for multiple activities” is one of the best practices when integrating UX into an Agile environment, since in this kind of environment, time is short to go deeper into the user research and evaluation [18].

Pattern: *Parallel Tracks*

Context: In agile environments concerned about UX or usability research, it is common to have a team in charge of UX-related activities, a team in charge of development activities and team leaders. In this context, one team provides the others with input through delivering results from user research, prototyping of new screens and interface evaluation, whereas the development team will share the code integrated to the interface delivered by the UX team. In this situation, there are parallel tracks involving the UX and development teams. Each team is responsible for its own activities, but communicating to each other during the cycle.

Problem: Fixing design flow in agile cycles, clarify doubts regarding design releases, get the design job done in one iteration, and validate designs. Main forces involved:

- **Force 1:** Performing UX activities synchronized with the features under development.
- **Force 2:** Implementation of the features already defined by UX research and design activities, and fix features based on UX evaluations.

Solution: UX or usability team working in a parallel track with the development team in order to synchronize their activities.

Example: The pattern follows the agile UCD cycle proposed by Sy [5]. Parallel tracks aims at synchronizing the UX activities with the development sprints [19].

Pattern: *UX Specialists as Full-Time Member of the Agile Team*

Context: Traditionally, UX or usability teams work on several projects at the same time, i.e. they are not fully dedicated to a unique project. This traditional UX styles of working does not fit into Agile development. In this case, UX specialists would start losing important updated data during the iterations. Moreover, UX professionals had not much time to dedicate themselves to each project and perform design activities. For these reasons, UX specialists started to be dedicated to one project at a time, reinforcing that all members are fully committed to the systems quality.

Problem: The UX team feels left out of the cycle or loses crucial information, with short time for design activities. Main forces involved:

- **Force 1:** In agile methods, all the participants must be committed and not just involved to the project. Daily communication and co-location are important.
- **Force 2:** UX teams might work with several projects at the same time. Thus, the communication between the UX team and the development team becomes even more difficult.

Solution: Participants working full-time in a single agile team. UX specialists should not work at multiple projects concurrently. Moreover, the pair designing practice – as in pair programming – is strongly recommended.

Example: Nielsen describes the practice named “Decline of the centralized UX department” as a result of his experiences. He says that all his case studies indicate that UX personnel should be co-located with developers and other members of the project. Thus, the UX team would be considered part of the project team [20].

4.2 Specify Context of Use

Pattern: *Little Design Up Front*

Context: The design upfront as proposed in traditional UX research does not fit into the agile values. Regarding agile values, requirements changes are considered inherent to the development process and it is not necessary to spend much time with planning and designing of requirements upfront. Due to these values, agile development advocates little design upfront the development, then just the necessary work is done, reducing waste. In this context, UX specialists focus only on the job related to those features for a specific iteration. Then, they share their results with the development team, according to the Parallel Tracks pattern definition.

Problem: Traditionally, user research is carried out before development starts. This is incompatible with agile values. Main forces involved:

- **Force 1:** In agile methods, only the features that will fit into the next iteration need to be detailed.
- **Force 2:** User research and the analysis of collected data that result in requirements definition are usually performed ahead of the start of the development.

Solution: Carrying out research with less granularity of details, focusing only on a set of essential features for the next iteration, i.e., chunking design work into small pieces.

Example: Cho [16] states that less time should be spent with high-fidelity design. The focus should be on defining problems and finding solutions collaboratively. Before starting the sprint cycle, the UX specialists team gets ready for the next sprint through holding sessions to define problems and develop solutions and this happens during the last week of the previous sprint.

Pattern: *Contact Plan of Users*

Context: Conducting research with real users depends on knowing who the typical users of a system are and then finding users willing to participate in research, design and testing stages. Additionally, in order to have these activities done before the next iteration, it is not possible to think about which users should be called. Having a contact plan of users is helpful in such a situation. Contact Plan of Users consists of a list of users who meet the systems profile, their contact data, and possibilities of participation are kept up-to-date.

Problem: Being unable to reach the right users in time, not getting feedback in time for the project, depending on beta-users, and delivering a design without any evaluation. Main forces involved:

- **Force 1:** The need for typical users to participate in activities such as requirements gathering, design and evaluation.
- **Force 2:** Check if the users are available to participate in UX activities is in accordance to project schedule.

Solution: Create an early channel for communication with final users in order to guarantee stable feedback.

Example: In [21] report constant planning of future feedback from users during development. According to the authors, the UX practitioner may does not know exactly what will be asked in two weeks, it could be a feedback for a prototype or several questions and answers involving typical users. However, it is necessary to schedule meetings with users on a regular basis to obtain feedback quickly.

4.3 Specify Requirements

Pattern: *User Stories*

Context: In XP, user stories are written by customers such requirements the system needs to meet for them. They are similar to case scenarios but they do not have only the working description of a user interface. User stories are

the basis for creating acceptance tests. The aim is to check if the user story was implemented correctly. The aim of writing user stories in agile teams is to meet user needs, not to specify user interfaces. However, when UX activities are integrated into agile methods, besides adding the real users point of view of the system and not only the customers – focused more on business requirements –, it is also common to include interface drafts into user stories. These drafts aim at improve the understanding of ideas previously discussed and creation of prototypes. Also, personas are used to better describe the user profile present in each story and usability issues can be inserted into the story acceptance criteria.

Problem: Traditionally, usability requirements are defined and described in detailed documents which contain task analysis, personas, an overview and case scenarios. However, extensive documentation does not fit into agile methods scenarios. Main forces involved:

- **Force 1:** In agile methods, extensive documentation is considered as a time consuming activity that can be easily outdated due to requirements changes.
- **Force 2:** Requirements definition with UX methods or usability usually describe in details user needs according to the results of user profiles.

Solution: Inclusion of usability requirements into user stories, defining acceptance criteria for each story.

Example: Hussain *et al.* state that user research can be used to develop user stories [22]. Sohaib and Kahn report that user stories must be integrated into design based on scenarios [23].

Pattern: *More Collaboration, Less Documents*

Context: More collaboration, less documentation refers to values that have been already shared in agile environments, but expanded to UX activities or usability. In this context, sharing research findings happens orally and through low-fidelity prototypes, such as paper drawings made with pencils in a way that the ideas are easily and quickly shared and validated. Also, in these cases, low and medium-fidelity prototypes are used as specification along with user stories, which share the same value – just the enough for understanding.

Problem: Non-understandable designs, designs left behind during implementation, agile team does not understand UX activities. Main forces involved:

- **Force 1:** Research findings and evaluations need to be shared among the team as soon as possible in the development process.
- **Force 2:** More detailed descriptions of the UX research findings or usability last longer because of the process of documentation.

Solution: Design and design processes being frequently updated; continuous communication during implementation and testing. Also, some teams use UX professionals as design facilitators who collect information from the people involved, such as users, customers and developers, sharing them among all project members.

Example: Six [24] informs that UX practitioners need to adapt their deliveries because when someone are working with the rest of the team, one can answer their questions directly rather than writing a style guide. Instead of writting a 90-page report, problems from a usability test can be immediately adress. Less time authoring unnecessary documentation means more time for you to help make a great product. The author also highlights the fact that since UX practitioners are much more involved with tasks outside their immediate specialty, such as development and testing, they find new ways of using their UX knowledge.

Pattern: *Prototypes as Specification*

Context: As part of the idea previously described in the pattern “More collaboration, less documents”, several teams use prototypes as specification rather than extensive documents aiming at gathering users and customers requirements for discussion as soon as possible. In this context, prototypes are used to quickly communicate and validate ideas related to user interface requirements. Usually, they are built by UX specialists and along with user stories they are the main sources for specifying system requirements.

Problem: Sharing interface requirements that meet user needs faster than documents of requirements specifications. Mains forces involved:

- **Force 1:** To spread knowledge related to user needs, which need to be met by the system in a faster way and early in the development process.
- **Force 2:** Need for registering requirements defined and shared with the team.

Solution: Use prototypes to specify user interface requirements.

Example: In [25], user interface prototypes are used to make customer requirements be known as soon as possible during discussions and make them work as a template for development.

5 Conclusion

The problem of combining User-Centered Design and agile methods is an example of a context-dependent issue. We believe that presenting usability practices as patterns – providing context, problem, solution, and example – will increase their applicability and improve their reliability. However, this is an ongoing work and it still needs further study in order to validate the proposed patterns.

Usability should be part of the development process, not only with the aim of creating usable products, but also usable practices, involving all team members related to the context in which the systems will be used. Thus, an agile and usability-based team will have all members focused on meeting the customers and typical users needs, since all of them are required to understand the importance of doing so.

All members participating in the development of a system are users of methods or practices in software development. When it comes to choosing the best method

in software development, it is necessary to consider who the users are, what they need, how and in which context they perform their tasks in order to focus on the human beings involved in the project. In other words, apply to the user of development methods the same tasks used to understand the end users of a system.

It is not necessary to use methods with great formality, regarding their application or documentation. However, it is necessary to have a collaborative environment, where all the people involved with the project share both the knowledge acquired from close contact with real users and the knowledge required to develop the product, starting from the data collection and specification of the context of use to the development and testing.

In this context, each project member benefits from the previous knowledge of the others, allowing a greater comprehension of the software development project as a whole. Also, it is possible to improve the usability of the product and the development process.

Patterns can help in understanding how to integrate the practices of the two communities involved and allow us to visualize the commonalities shared. The goal of the proposed patterns is to facilitate the use of the best agile usability practices by identifying more clearly in which context the pattern can be applied, and what is the problem that each pattern solves, presenting examples.

References

- [1] Dingsøyr, T., Dybå, T., Moe, N.B.: *Agile Software Development - Current Research and Future Directions*, 1st edn. Springer (2010)
- [2] Ratcliffe, L., McNeill, M.: *Agile Experience Design: A Digital Designer's Guide to Agile, Lean, and Continuous*. New Riders (2012)
- [3] Larman, C.: *Agile and Iterative Development: A Manager's Guide*. Pearson Education (2003)
- [4] Association, U.E.P.: *What is user-centered design?* (January 2014)
- [5] Sy, D.: Adapting usability investigations for agile user-centered design. *Journal of Usability Studies* 2(3), 112–132 (2007)
- [6] Fox, D., Sillito, J., Maurer, F.: Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In: *Agile 2008 Conference*, pp. 63–72 (2008)
- [7] Silva da Silva, T., Martin, A., Maurer, F., Silveira, M.: User-centered design and agile methods: A systematic review. In: Society, I.C. (ed.) *Agile Conference, Agile 2011*, pp. 77–86 (2011)
- [8] Santos, A.P.O.: *Application of agile usability practices in free and open source software*. 131 p. thesis (master in computer science) (2012); Institute of Mathematics and Statistics, University of Sao Paulo, Sao Paulo, <http://www.teses.usp.br/teses/disponiveis/45/45134/tde-22082012-154721>
- [9] Schwaber, K., Beedle, M.: *Agile Software Development with Scrum*, 1st edn. Prentice Hall PTR, Upper Saddle River (2001)
- [10] Beck, K., Andres, C.: *Extreme Programming Explained: Embrace Change*, 2nd edn. Addison-Wesley Professional (2004)

- [11] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., King, I.F., Angel, S.: A Pattern Language: Towns, Buildings, Construction. Center for Environmental Structure Series. Oxford University Press (1977)
- [12] Tullis, T., Albert, B.: Measuring the User Experience: Collecting, Analysing, and Presenting Usability Metrics. Morgan Kaufmann (March 2008)
- [13] Chamberlain, S., Sharp, H., Maiden, N.: Towards a framework for integrating agile development and user-centred design. In: Abrahamsson, P., Marchesi, M., Succi, G. (eds.) XP 2006. LNCS, vol. 4044, pp. 143–153. Springer, Heidelberg (2006)
- [14] Belchev, B., Baker, P.: Improving obama campaign software: Learning from users. In: Agile 2009 Conference, pp. 395–399 (2009)
- [15] Sy, D., Miller, L.: Optimizing agile user-centred design. In: CHI 2008 Extended Abstracts on Human Factors in Computing Systems, pp. 3897–3900. ACM, New York (2008)
- [16] Cho, L.: Adopting an agile culture. In: Proceedings of the 2009 Agile Conference, AGILE 2009, pp. 400–403. IEEE Computer Society, Washington, DC (2009)
- [17] Gothelf, J.: Ux designer as product owner (February 2011), <http://www.jeffgothelf.com/blog/ux-designer-as-product-owner/>
- [18] Patton, J.: Emerging best agile ux practice (2008), http://agileproductdesign.com/blog/emerging_best_agile_ux_practice.html
- [19] Lu, C., Rauch, T., Miller, L.: Agile teams: Best practices for agile development 9(1), 6–10 (2010)
- [20] Nielsen, J.: Agile user experience projects (2009), <http://www.useit.com/alertbox/agile-user-experience.html>
- [21] Enterprise, E.: Incorporating user-centered design into an agile development process (December 2011), <http://experioinc.com/incorporating-user-centered-design-into-an-agile-development-process/> (accessed on: December 2011)
- [22] Hussain, Z., Milchrahm, H., Shahzad, S., Slany, W., Tscheligi, M., Wolkerstorfer, P.: Integration of extreme programming and user-centered design: Lessons learned. Agile Processes in Software Engineering and Extreme Programming 31, 174–179 (2009)
- [23] Sohaib, O., Khan, K.: Integrating usability engineering and agile software development: A literature review. In: 2010 International Conference on Computer Design and Applications ICCDA, vol. 2 (2010)
- [24] Six, J.M.: Integrating ux into agile development (April 2011), <http://www.uxmatters.com/mt/archives/2011/04/integrating-ux-into-agile-development.php> (accessed on: December 2011)
- [25] Broschinsky, D., Baker, L.: Using persona with xp at landesk software, an avocent company. In: Proceedings of the Agile 2008, pp. 543–548. IEEE Computer Society, Washington, DC (2008)