Integrating Extreme Programming and User-Centered Design [⋆]

Zahid Hussain[1], Martin Lechner[1], Harald Milchrahm[1], Sara Shahzad[1], Wolfgang Slany[1], Martin Umgeher[1], and Peter Wolkerstorfer[2]

[1] Institute for Software Technology, Technical University Graz, Austria,
m3@ist.tugraz.at
[2] CURE - Center for Usability Research & Engineering
wolkerstorfer@cure.at

Abstract. The success of a software development project is associated not only with tools and technologies, but it also depends on how much the development process helps to be user-centered and developer-oriented. Involving customers in the process and being people-oriented, Extreme Programming (XP)– One of the popular agile methods – can be a choice for developing a usable system. The project under study is a multimedia streaming application for mobile phones that allows to make content-based search for audio and video content in large databases and play it on a mobile phone virtually anywhere, at any time. Our approach to application development focuses on the adoption of XP and User-Centered Design (UCD), emphasizing iterative user-interface development involving usability engineers and end-users.

The paper describes the process of integrating XP with user-centered design and shows how an agile development technique facilitates to be user-oriented and at the same time preserves the social values of the development team.

1 Introduction

"An inherently usable and technically elegant application cannot be considered a success if it does not satisfy the end-users' needs. End-users are often left out of the development process" [9]. A usable software application should focus on its end-users, theirs goals, and their satisfaction. Agile development processes - especially XP - involve customer as a business representative who is responsible to specify the business value of user requirements and prioritize them accordingly in the development. Along with this, XP possesses all the advantages of: on-time delivery, optimized resource investments, short release cycles, working high quality software, tight customer integration, incremental design and test driven development [1] [2] which are all in the favor of customer and ultimately benefit the end-user. Also, being people oriented it defines the whole social structure which is needed to run a development process in a productive way.

UCD is a design approach focused on the information about the people who are the actual users of the product. This user focus is maintained by considering this information during planning, design and development of a product [13].

Although XP and UCD are two different methodologies but both focus on the user. Due to this same main focus both methodologies can be integrated very easily [3]. The integration will obviously result in complementing each other and resulting process will allow us to gain the advantages of both worlds and at the same time minimize the deficiencies of both methodologies. As, XP lacks in knowing their true users and UCD lacks of a flexible and adaptive development methodology that lasts throughout the entire project [12].

Here in the context of this project we integrate XP and UCD [4], utilizing different HCI (Human Computer Interaction) instruments like user studies, personas, usability expert evaluations, usability tests, and automated usability evaluations [5].

The following section is meant to explain the points of integration in both methodologies by comparing the values that the two methodologies possess. Then we explain our project and

[⋆] The research herein is partially conducted within the competence network Softnet Austria (www.soft-net.at) and funded by the Austrian Federal Ministry of Economics (bm:wa), the province of Styria, the Steirische Wirtschaftsförderungsgesellschaft mbH. (SFG), and the city of Vienna in terms of the center for innovation and technology (ZIT).

team setting to show in which context and by whom the process is used. We proceed by going into the details of our UCD process and also outline the results of a usability study recently conducted by the usability engineers. Finally, the conclusion is given.

2 Common Values of XP and UCD

The core values of XP and UCD [4] are applied to solve different issues: In XP, a simple implementation, fulfilling the minimum requirements of the application, is created and iteratively extended, while UCD tries to continuously improve the usability of the user interfaces. However, when comparing some of the core values, it seems obvious that the two development processes can benefit from each other's practices.

2.1 End-User Involvement

One of the core practices of XP is to have an *On-Site Customer*, a real user of the application under development who is co-located with the programmers in order to answer domain-specific questions and give feedback on the system. This practice matches well with the testing of prototypes with actual users as proposed by UCD.

2.2 Testing

Constant and extensive testing is at the heart of XP. It is mainly embodied by two practices: *Continuous integration* runs all existing automated tests whenever the code base is changed or extended in order to check if the changes caused any undesired side effects. Most of these tests emerge from *test-driven development*: first, automated tests checking the desired behavior are created; then, the actual behavior is implemented and can right away be evaluated with the tests. This usually is done only for pure behavioral code but can be extended to user interfaces: tests can check the expected behavior of an interface, and these tests can be run whenever the code is changed.

The end-user tests of UCD are a valuable source for test targets: an unexpected user action that caused a problem in the application can be replicated as an automated and continuously evaluated test to ensure that the problem, after solving it once, does not reappear.

2.3 Iterative Development

Both XP and UCD propagate an iterative procedure [4] [1] of design and development [13]. An XP project yields *small releases* (another core XP practice) on a regular and frequent base (usually a few months). Each release version is based on the previous one, incorporating new features and fixing bugs of the predecessor. Inside a release time frame, work is organized in "iterations" (usually taking one to four weeks). On an even smaller scope, many feedback-and-change iterations take place, especially in conjunction with test-first development and *refactoring* (the practice of changing source code in order to improve its quality without changing its functionality).

UCD also proposes a design–test–modify circle for developing user interfaces. The scope of iterative development in XP and UCD differs: releases and iterations in XP are mainly organizational units, while refactorings are just considered a development tool; on the other hand, UCD's iterative user interface refinement is a more explicit process as its involvement of external persons (the test users) makes it more complex. Nonetheless, iterative interface development of UCD fits well into the iteration principle of XP because both approaches are aware of the value (and necessity) of evolutionary development.

3 Project and Team Setup

We are developing an application that enables a user to perform content-based search for audio/video content and play it on a mobile phone. This content includes radio and TV archive material, like documentaries or other recordings of historical, political and cultural importance, discussion programs, movies, music videos, audio books, and music. The application is being designed keeping in mind the social interaction of users. The system provides different community-building features to encourage interaction amongst them [5].

In addition to this, one goal of the project is the analysis of agile software development methods, particularly XP, and to devise a usability test procedure for mass applications on mobile devices with emphasis on UCD and iterative user-interface design.

The team consists of six full-time regular members, having different social and cultural background; five developers (two of them are from South Asia and the others are from Europe)and a product manager who plays the role of the *On-Site Customer*, allowing us to implement this XP core practice.

The customer communicates with the project partners who come from various domains, including UI design, usability research, telecommunication, content providing, and hardware infrastructure. Developers also directly communicate with the engineers of a partner usability research center regarding usability issues. The usability engineers working for our project are active in UCD research with the team.

4 The Design Process

The following sections describe process which is followed for the User-Centered Design of our application.

4.1 Approach to User-Centered Design

User-interface design plays very important role in the acceptance of a web based application by the users. The overall process of our approach to UCD is based on evaluating the usability of the application in small iterative steps. This helps us to gain insight into the real users' functional and cognitive requirements. We design prototypes of the user-interface of the system and test them throughout the development process. As a result the fidelity of the prototypes increases and evolves.

The work flow presented in Figure 1 illustrates the iterative design approach to incorporate UCD into our XP process. From a broad perspective, the application development cycle starts from defining the user stories (user-required application features), then comes to mock-up designing and at the end actual implementation is performed. The process is performed as follows:

- Different feature-related user stories of the application are created by the customer in coordination with all the stakeholders.
- Developers create different paper mock-ups for each of the required features to collect ideas and to present them to the customer.
- The customer decides which one of the mock-ups best suits his needs, or he suggests any modifications to the mock-ups.
- A final mock-up is derived according to customer's likeness which then serves as the basis for actual implementation.
- Once implementation mock-up of a feature (or group of related features) is finished, the usability engineers are asked to give the feedback on it.
- After incorporating the feedback given by the usability engineers into the application the end-user tests are conducted on the application by the usability engineering team.

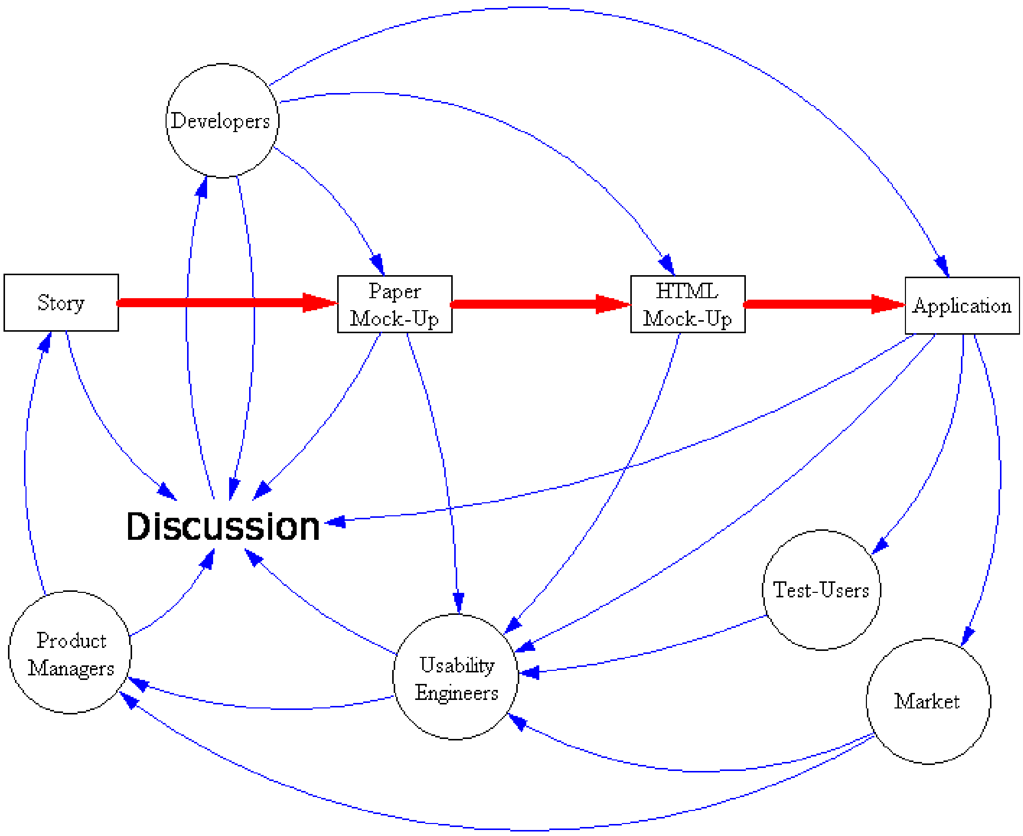

Fig. 1. Iterative UI design workflow [5].

- The feedback on the application from the usability engineers, as well as, from the test-users is taken as an input for further refinements in the UI design of the application.
- The results are then incorporated into automated tests, by employing test driven development, which are used as an executable specification for the actual implementation.

This feedback-and-change cycle provides insights into whether the user-interface design is meeting the usability criteria. As the application development is done in short iterations, the developers are able to refactor the system continuously according to the feedback derived from the parallel, as well as iterative, UI design process. Hence, the system evolves according to the needs of the end-users and the specifications derived from actual usage.

4.2 Choosing the Type of Mock-up

We make use of two different types of mock-ups; low fidelity paper mock-ups and high fidelity implementation mock-ups. The benefit of using paper mock-ups for the interaction design is that they can be designed and modified quickly. For simple interaction designs, a low fidelity paper mock-up suffices as a basis for further discussions and the implementation. An additional advantage is that it is easier to criticize simple and rough mock-ups compared to ones which look neat and perfect from the graphic design perspective [10]. But for some features a high fidelity mock-up is required to clearly visualize the interface. As we have the benefit of an on-site customer co-located with the development team all the time, so for those tasks a quick implementation mock-up is designed and shown to the customer. That implementation mock-up is then modified accordingly if required by the customer. If our customer was not co-located with us all the time then it would have been difficult to take maximum benefit out of this quick feedback-and-change cycle.

4.3 Frequency of End-User Tests

The end-user tests are made on on-demand basis. That is, when customer says that now is the appropriate time, from the business point of view, to run a usability test with test-users. Also, when there is enough amount of new functionality added to the application it becomes effective to perform the usability tests and then do further development. It would have been wonderful if user tests could be made on regular basis, like at the end of each release, but considering the expenses and resources required for it we have kept it on only on-demand basis. Therefore, the expensive part of involving real users is done more effectively.

4.4 The Testing Workflow

Figure 2 describes the model of the usability engineers integrating the HCI instruments (user studies, personas, extended unit tests, usability tests and usability expert evaluations) into the XP process [14]. It shows the interplay of the HCI instruments into the XP process. Applied correctly in different phases of the project the instruments are designed to reach the goal of improved usability of the application. It can be seen that end-users are integrated in two different ways: on one side user studies are taken into account to develop personas [7] which then specify direction of development (by guiding customer to identify user stories), and also at the end of a development iteration the vision about the users is broadened which helps in extending the personas. This serves as an indirect end-user input for the development process. On the other side feedback from usability tests performed by test-users (as part of the usability evaluations) serves as a direct input for further enhancement and development of the application [14].

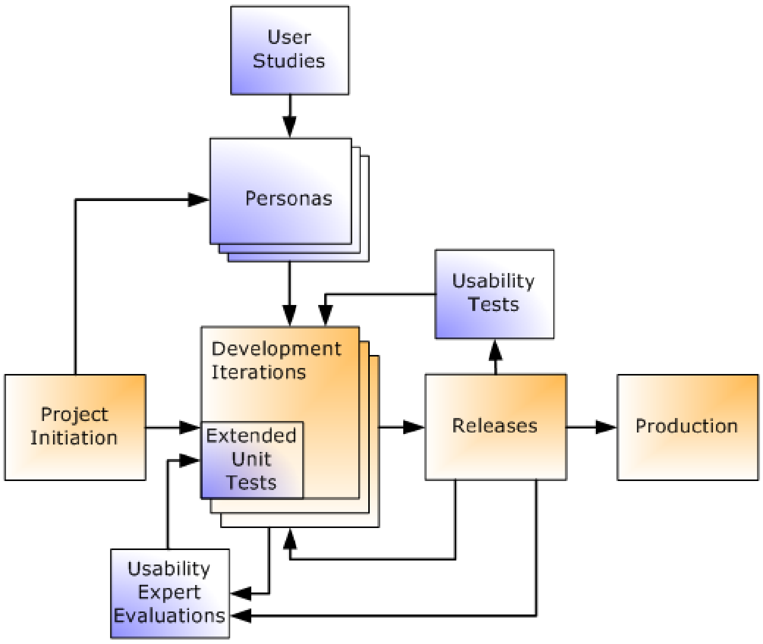

Fig. 2. The integration of HCI instruments into XP [14].

4.5 Feedback from a Usability Test

The following points were highlighted as the result of a recent usability study conducted by the usability engineers on the latest version of the application available at that time. Figure 3 shows a screen shot of the application. The application was evaluated with ten test-users on a specific mobile phone.

- Improvements of layout and design. Some major changes were suggested in the layout and design of the user-interface by the usability engineers according to the feedback given by the test users.
- Improvements of the prototype's usability. Some usability issues became apparent for some navigation and data controls on the interface
- From the mock-ups of three schemes. The users were also presented three different color schemes of the application to have a view about the over all aesthetic design of the application. They graded the color schemes as best average and worst.

During this study, two of the developers were participating as observers. This provided a great opportunity to the developers to see how users actually reacted to simple items and controls used in the interface and imagined their usage. At the end of the study, many new stories were generated from the observations and interviews with the end-users. This approach of testing involves the end-users directly in eliciting their own requirements: how they perceive the application, and what they want from the application.

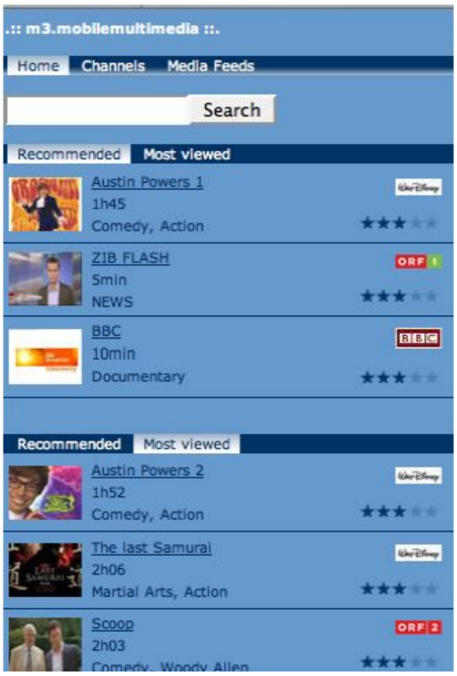

Fig. 3. The prototype of the home page

4.6 Testing Issues

A big issue in mobile user-interface practice is that current approaches are not sufficient for mobile phones [11]. For designing any software use of good UCD practices ensure that the product works [8]. This further supports the use of UCD approach for UI design. To enhance it further, we provide high fidelity implementation prototypes to our usability engineers for user testing. As paper prototypes are good and sufficient for verifying non mobile-based product requirements, but in case of applications for mobile phones they are not sufficient for finding out and solving usability issues related to detailed interaction [8]. Also, this is very important that the application is tested on mobile phone and not on some web based simulator for understanding the interaction issues concerning the use of mobile phone interface [8].

5 Conclusion

XP is a lightweight process that puts very little administrative overhead on the developers. Therefore, extending XP with additional practices is much easier than for other, more restrictive methodologies. The integration of usability engineering methods works especially well because of the many overlapping principles (e.g. iterative development, end-user incorporation) of XP and UCD.

The UI design process according to UCD is largely beneficial as it provides feedback [3] which is used for the system's functional requirements. The assessment of each feature from the users' perspectives influences the whole development process of the application and addresses the problems which arise when the system requirements are gathered only by discussions with stakeholders [6].

When deciding about usability issues in our project, we try to involve not only the development team and the product manager, but also the usability engineers and all project stakeholders, especially end-users. This practice led to an application that from the beginning was lacking many of the teething troubles common to technician-dominated development teams and can be seen as a big success factor for our project.

References

1. Kent Beck. *Extreme Programming Explained: Embrace Change.* Addison-Wesley Professional, 1st edition, October 1999.
2. Kent Beck and Cynthia Andres. *Extreme Programming Explained : Embrace Change (2nd Edition).* Addison-Wesley Professional, November 2004.
3. Bengt Göransson, Jan Gulliksen, and Inger Boivie. The usability design process - integrating user-centered systems design in the software development process. *Software Process: Improvement and Practice*, 8(2):111–131, 2003.
4. Jan Gulliksen, Bengt Göransson, Inger Boivie, Stefan Blomkvist, Jenny Persson, and Åsa Cajander. Key principles for user-centred systems design. *Behaviour & Information Technology, Special Section on Designing IT for Healthy Work.*, Vol. 22 No. 6:Pages 397–409, 2003.
5. Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Martin Umgeher, Thomas Vlk, and Peter Wolkerstorfer. User interface design for a mobile multimedia application: An iterative approach. In *ACHI 2008, First International Conference on Advances in Computer-Human Interaction, February 10-15, 2008, Sainte Luce, Martinique, France*, pages 189–194. IEEE Computer Society.
6. Timo Jokela and Pekka Abrahamsson. Usability assessment of an extreme programming project: Close cooperation with the customer does not equal to good usability. In *5th International Conference, PROFES '04*, pages 393–407, 2004.
7. Plinio Thomaz Aquino Junior and Lucia Vilela Leite Filgueiras. User modeling with personas. In *CLIHC '05: Proceedings of the 2005 Latin American conference on Human-computer interaction*, pages 277–282, New York, NY, USA, 2005. ACM.
8. Eeva Kangas and Timo Kinnunen. Applying user-centered design to mobile application development. *Commun. ACM*, 48(7):55–59, 2005.
9. Marc McNeill. User centred design in agile application development. http://www.thoughtworks.com/pdfs/agile_and_UCD_MM.pdf.
10. Reinhard Sefelin, Manfred Tscheligi, and Verena Giller. Paper prototyping - what is it good for?: a comparison of paper- and computer-based low-fidelity prototyping. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 778–779, New York, NY, USA, 2003. ACM Press.
11. S.R. Subramanya and Byung K. Yi. User interfaces for mobile content. *IEEE Computer*, 39(4):85–87, April 2006.
12. Anders Toxboe. Introducing user-centered design to extreme programming. May 2005. http://blog.anderstoxboe.com/uploads/16082005_XP_and_UCD.pdf.
13. W3C. Notes on user centred design process (ucd). http://www.w3.org/WAI/EO/2003/ucd, April 2004. Last visited: 18.08.2008.
14. Peter Wolkerstorfer, Manfred Tscheligi, Reinhard Sefelin, Harald Milchrahm, Zahid Hussain, Martin Lechner, and Sara Shahzad. Probing an agile usability process. In *CHI '08: CHI '08 extended abstracts on human factors in computing systems*, pages 2151–2158, New York, NY, USA, 2008. ACM.