Probing an Agile Usability Process

Abstract
In this paper we describe adaptations to the classical Extreme Programming (XP) process. The approach described integrates HCI (human computer interaction) instruments. The implemented HCI instruments are: user studies, extreme personas (a variation of the personas approach), usability expert evaluations, usability tests, and automated usability evaluations. By combining XP and UCD (user centered development) processes we take advantages of both approaches.

Keywords
Usability Engineering Process, User Centered Development, Extreme Programming, Personas, Usability Engineering Tools, Mobile Devices

ACM Classification Keywords
H5.2 User Interfaces: User-centered design, prototyping, screen design, graphical user interface (GUI), evaluation/methodology.

Industry/category
Web application, Mobile application, eMedia, Telecom, Entertainment

Introduction
This case study sums up the setup of an adapted Extreme Programming (XP) process.
The goal of the adaptation was to examine the applicability of a usability-aware agile software development process. The process was designed to allow applying HCI instruments in XP teams. This should combine the advantages of the Extreme Programming methodology (on-time delivering, optimized resource investments, short release cycles, working high quality software, tight customer integration) with the advantages of a user centered design process (usable, accessible, and accepted products, end-user integration).

The context where these instruments are utilized is a project where we develop a mobile multimedia application. The application enables users to create customized mobile multimedia channels based on keywords. The keywords are matched against metadata and transcriptions. Here an example: it is possible to create a news channel with all news-broadcasts mentioning "formula one".

**Problem Statement**
Experts doubt that the XP process leads to true user-centered design [6]. Following issues can prevent the integration of HCI instruments into the XP processes:

**Ad-hoc Input**
Because of the short release cycles software engineers would need ad-hoc usability input during the software development. In practice usability input is not given ad-hoc, but after longer periods (in the average between one to two weeks average). XP practitioners cannot accept such time-spans.

**Patchwork experience**
Our work-experience suggests that users often experience software as patchwork when developed from bottom up like done in XP. We attribute this to the missing holistic view at the beginnings of XP projects. The programming activities are not based on a design concept.

**Cultures**
Another problem is the difference between cultures: Software engineers on the one hand and HCI experts on the other hand come from different domains with different attitudes, approaches, backgrounds, and even different ways to express themselves. The XP process requires tight cooperation in teams, which reveals differences between engineers and HCI experts very quickly: engineers have a technical approach to software development whereas HCI experts mainly have a psychological background, hence taking a cognitive view on the software development. These differences can lead to problems. Methods to prevent this have to be integrated into the collaboration process.

**Technical Focus**
Unit tests in XP environments are designed for technical testing. Hence the focus is on technical functionality – ignoring usability issues. This means that the technical view of testing has to be expanded by HCI approaches and means.

**On-site Customer Representative**
From an HCI point of view the inclusion of customers is a step into the right direction. But: the Manifesto for Agile Software Development [11] does not clearly demand end!users as customers. We expect deficits in...
usability if it is not clearly stated that end-users have to be part of the process. Developers need to have a clear picture of the end-users.

**Background**
Creating usable high quality software is not trivial – especially when it comes to mobile applications. Different processes and tools are already in place to create high quality and usable software. They come from different domains (engineering, design, HCI). The processes and tools we chose from are:

- Extreme Programming: the agile software development process
- Extreme programming and the user centered development (UCD) process: the combination of software engineering and HCI skills
- Personas: a tool to define end-user groups by archetypes to raise empathy for end-users in software development teams; we extend the persona approach to “extreme personas” to fit the changing requirements appearing during the project duration
- Automated usability evaluation as a tool and the technical “glue” between HCI knowledge and the XP unit tests. We want to extend known code-based usability evaluation by semantics to create (semi-) automated test-scripts which can be included into the set of technical unit tests; these extended unit tests allow HCI experts to directly apply HCI knowledge in the process

**Extreme Programming (XP)**
XP is an agile software development process. The goal of this process is to deliver high-quality software in time. This is done by different means: test driven development (by using unit tests), short iteration cycles, on-site customers, pair-programming, refactoring (restructuring an existing body of code, altering its internal structure without changing its external behavior), and user stories (where requirements are captured in short narrative stories) are the most important of them; tc. [9, 10]. Figure 1 shows the basic XP process where we will build our adapted approach on.

**Figure 1.** Agile development process; the release cycles are three month, the iteration cycles are one week

**Extreme Programming and UCD**
Practitioners already combined UCD (user centric development) and XP by varying approaches ([5], [14] and [2]). McInerney, P. and Maurer [12] showed already that a marriage of these two approaches is possible [12]. In our study we investigated the prerequisites and the process of such an integration.
**Personas**

The personas method was developed as a tool for raising empathy for the end users in development teams, and as a means for communicating peer group definitions. When developing personas we design archetypical prototypes of end-users. This is done by accumulating knowledge about intended peer groups. One persona represents a typical user group. A persona—as an archetypical figure—can guide decisions about product features, navigation, interactions, and even visual design (among other factors) [13].

**Automated Usability Evaluation (AUE)**

The idea of automated usability evaluation is not new. Basic research goes back to the early nineties [4, 1]. In the year 2000 the state of AUE is still described as "quite unexplored" [7]. Current approaches tend to focus on multimodality [15] and mobile devices [17]. Besides their limited scope (because most tools evaluate on a code-basis) for our project the existing tools have a big disadvantage: most of them are isolated solutions for HCI experts. Hence they hardly integrate seamlessly into the existing development processes.

**Agile Usability Process**

**General Approach**

The novelty of our approach is that we do not rely on one or two selected instruments but took five of them and integrated them into the XP process. Figure 2 shows the interplay of the HCI instruments related to the XP process. Applied correctly in different phases of the project the instruments are designed to reach the goal of maximized software quality (in terms of technical quality but also in terms of usability).

**Figure 2.** This is the modified agile development process with usability instruments included (User studies, personas, extended unit tests, usability tests and usability expert evaluations). It can be seen that end-users are integrated in two different ways: on the one hand user studies inform the development and extension of the personas – which gives indirect end-user input to the developers, on the other hand usability tests (as part of the usability evaluations) directly inform development.

The idea was to integrate these five instruments into the classical XP process. This multi-instrument approach was developed to solve all the problems lined out in the problem statement above.
The five HCI instruments we rely on are:

- Extended unit tests for automated usability evaluation
- Extreme Personas (a variation of the classical personas method) extend the typical XP user stories
- User studies (focus groups, diaries, laddering-interviews) are not inherently foreseen in the XP process and are integrated to extend the XP concept of the on-site customer
- Usability expert evaluations to solve the ad-hoc input problem
- Usability tests to solve the problem of the on-site customer representative

**Extended Unit Tests**

In XP unit testing is mandatory. Our approach extends the technical unit tests by adding usability-specific test cases. Code based tests are enhanced with semantics to achieve this goal – for example: code based tests can check against guidelines like the usage of capital letters on buttons. When adding semantics (the correct label of a button) we can include the test into the set of unit tests already used in XP. Test-driven development in XP means to write tests first. The written tests then define the behavior of the application. Adding usability related unit tests with semantics, allows us to define the usability of the application. Unit tests – by definition – test small definable units of the software. The problem of patchwork application suggests using a holistic approach for testing. Therefore the unit tests are extended by tests, which go beyond single units, and test complete interaction flows.

**Extreme Personas**

This approach starts with the same activities like the classical persona method: preliminary user groups are defined and personas are modeled for them afterwards. During user studies new knowledge leads to two distinct actions: when the new knowledge suggests slight changes for a persona the persona will be refactored, if the found knowledge reveals that current personas do not cover the new insights new personas will be developed. These actions make the classical personas “extreme” by applying the XP paradigm of small iterative steps and refactoring – which is extending the personas in this case. During the coding phases the developers pin the personas beside the user stories. Their first application is in the planning games (the phase where user-stories are created) where the extreme personas yield as reference representation of the on-site customer.

**User Studies**

User studies are the instrument for getting knowledge about the end-users. The outcome of the user studies informs the design in two ways: on the one hand knowledge for creating and extending the personas is created; on the other hand direct input for the user stories can be derived.

**Usability Expert Evaluations**

Usability expert evaluations solve the problem of ad-hoc input. This is done by IM (instant messaging), email, and video-conferencing. Mock-ups (in early phases) and screens (in later phases) are sent to the HCI experts who then give ad-hoc input by using the channels mentioned above.
**Usability Tests**

Usability laboratory tests will include end-users as demanded by the UCD process but not demanded by the XP process (where it is not mandatory that end-users are part of the on-site customer representative).

**Interplay of Usability Instruments**

How do the single instruments work in practice? Based on the XP process HCI experts can intervene in three ways:

- Creation of user-stories (this was done by providing HCI knowledge – derived from studies, literature, and tests)
- Contribute by writing tests
- Extend XP methods (we brought extreme personas to the development team)

The knowledge derived from the used HCI instruments informed all of these activities.

**Lessons Learned**

We use the described process since summer 2007. The project will end in 2010. Then the final usability tests will prove if the process has been able to enhance usability of the applications. Until now we did not experience any cultural problems. The HCI experts in the project are well integrated into the development team.

The tight coupling of different expertise has led to a high motivation among project members. Developers gained insight into the subtleties of UCD, HCI experts learned to understand the origins of some of the usability problems.

Furthermore, we saw that the diverse technical testing frameworks demand technically aware HCI experts. Depending on the used frameworks the programming expertise required varies. In practice this could get a problem when the chosen framework is complex and little time for learning is available.

**Acknowledgements**

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).

**References**


