

## Practical Usability in XP Software Development Processes

Zahid Hussain<sup>\*‡</sup>, Martin Lechner<sup>†</sup>, Harald Milchrahm<sup>†</sup>, Sara Shahzad<sup>†</sup>, Wolfgang Slany<sup>†</sup>,  
Martin Umgeher<sup>†</sup>, Thomas Vlk<sup>†</sup>, Christina Köffel<sup>‡</sup>, Manfred Tscheligi<sup>‡</sup> and Peter Wolkerstorfer<sup>‡</sup>

<sup>\*</sup>Quaid-e-Awam University, Nawabshah, Pakistan

<sup>†</sup>Institute for Software Technology, Graz University of Technology, Austria

Email: {zhussain, hmilchra, sshahzad, wolfgang.slany, mumgeher, tvlk}@ist.tugraz.at

<sup>‡</sup>CURE – Center for Usability Research & Engineering, Vienna, Austria

Email: {koeffel, mtint, wolkerstorfer}@cure.at

**Abstract**—This paper describes the experiences made and lessons learned in an Extreme Programming (XP) software development project. We investigate the potential of XP to produce user experience-optimized products by including HCI experts in the team. We relate the software development method to user-centered design instruments and propose solutions to different user experience integration problems. Additionally, the practicability of different HCI instruments regarding solving those problems is examined. The analyzed instruments and methods are: user studies, personas, usability tests, user experience expert evaluations, and extended unit tests. The conclusion provides tips and tricks for practitioners.

**Keywords**-Agile Methods; Extreme Programming; Usability; User Experience; User-Centered Design.

### I. INTRODUCTION

As computers and mobile phones have become an essential part of everyone’s life, the target audience of software applications has shifted from technical experts to general consumers. Accordingly, the typical usage context of information and communication technologies (ICT) has moved from the office to the home. ICT is used for entertainment in a way that was not present in the days of office-only usage. Therefore, the user experience has become an increasingly important aspect of the adoption and the success of a software product. Today’s users expect powerful but nevertheless easy to use applications. Hence, user-centered design (UCD) processes are needed.

At the same time the pressure on the software market has increased. Shorter time-to-market as well as the constantly changing needs of the market and customers impose new demands to software development processes. Agile lightweight methods like Extreme Programming (XP) are designed to cope with these new conditions. For the development of a successful software product it is therefore inevitable to integrate user experience into an agile software development process.

In this paper, we provide background information and advice based on the experiences made in a research project with a practical application. The goal of the project was to increase the quality of software development processes.

Although we mainly discuss agile development processes (specifically XP), we also provide relevant information on how to include user experience techniques into other existing development processes.

### II. BACKGROUND

#### A. Software Development Processes

Software development processes are an attempt to structure and standardize development to make the outcome of a project plannable and predictable. There are many process models available and being used, ranging from heavyweight to agile. In accordance with the project circumstances the actual development process has to be tailored to the specific needs.

The choice of the right process depends on various factors: Risk level, requirements stability, time-to-market, etc. The clearer the requirements and the more stringent the organizational structure, the more structured and heavyweight the process has to be. The more unstable the requirements are, the more iterative and agile the underlying process has to be to cope with the changes. The shorter the time-to-market, the more lightweight and iterative the process should be to avoid administrative overhead and provide deployable software after each iteration [1].

#### B. Agile Software Development

The next logical step was the invention of agile software development methods. In software engineering, agile software development or agile methods refer to low-overhead methodologies that accept that software is difficult to control. They minimize risk by ensuring that software engineers focus on smaller units of work, business priorities, and high quality. One way in which agile software development is generally distinguished from “heavier”, more process-centric methodologies, for example the waterfall model, is by its emphasis on values and principles rather than on processes. Typical development cycles are one week to one month. At the end of each cycle the project priorities are re-evaluated. This is a feature that is shared with iterative

development methodologies and most modern theories of project management.

1) *Extreme Programming*: One well known method from the group of agile software development methods is XP. The XP methodology was formulated by Kent Beck, Ward Cunningham and Ron Jeffries. In March 1996 Kent Beck started a project at DaimlerChrysler using new concepts in software development [3]. The result was XP. The starting point was to find out what made software easy to create and what made it difficult. Kent Beck came to the conclusion that there are four factors to improve a software project: communication, simplicity, feedback, and courage are the values sought out by XP programmers [3]. In the second edition of his book, Beck added 'respect' as the fifth value [4].

### III. USER EXPERIENCE AND SOFTWARE DEVELOPMENT

The success of a software development project is not only associated with tools and technologies but also depends on how much the software development process helps to be user-centered and end-user-oriented [5].

#### A. User experience in XP/Agile Projects

Some experts doubt that the XP process leads to true user-centered design [6]. The issues arising from this problem statement suggest that XP and UCD would not fit. But this perception is simplistic and misguided as shown by successful applications in practice [7]. We can see succeeding practitioners combining user experience/UCD and XP/agile methods by varying approaches [5][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24].

1) *Why do some XP projects fail when including HCI work?*: The following issues can prevent the integration of HCI instruments into XP processes [17]:

- Ad-hoc Input: Because of the short release cycles software engineers would need ad-hoc user experience input during development. In practice, user experience input is not given ad-hoc, but after longer periods (one to two weeks average). Such time-spans are not acceptable for most XP practitioners.
- Cultural problems: 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 while communicating. 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 psychology backgrounds, hence taking a cognitive view on software development. As these differences can lead to problems, methods to prevent this have to be integrated into the collaboration process.

- Technical Focus: By its genesis unit tests in XP environments are designed for technical testing. Hence, the focus is on technical functionality – ignoring user experience 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 does not clearly demand end-users as customers [49]. We expect deficits in user experience 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 humans they develop for.
- Awareness: In order to successfully include user experience and user-centered design in an XP process the developers need to have a basic understanding of user experience issues.

### IV. CASE STUDIES

#### A. Setting

Mobile computing is leading a revolution. Studies show that multimedia – Audio and Video (AV) – consumption is on the edge to become one of the next killer applications for mobile devices [25]. Still, there are not many full-featured applications on the market which utilize the available bandwidth and are accepted by consumers.

We have been developing an application in this field that enables a user to perform content-based search for AV content in large digital archives and play it on a mobile phone. The major problem for an average user in this context is the combination of the overwhelming amount of multimedia content available and unsatisfactory user-interfaces for accessing it. User experience is the key success factor for such applications [5][26].

#### B. Approach

The novelty of our approach lies in the fact that we do not only use one or two methods related to user experience to integrate them into the XP process but selected five instruments to enhance the existing XP process. This multi-instrument approach was developed to solve the problems as introduced in section III-A. Applied correctly in different phases of the project these instruments are designed to reach the goal of maximized software quality in terms of technical quality and also in terms of user experience [5][17][18]. The five instruments we relied on are:

- Extreme Personas
- User Studies
- Usability Tests
- User Experience Expert Evaluations
- Extended Unit Tests

The integration of these HCI instruments into the XP process is shown in Figure 1. The following sections will provide a short overview of the single methods.

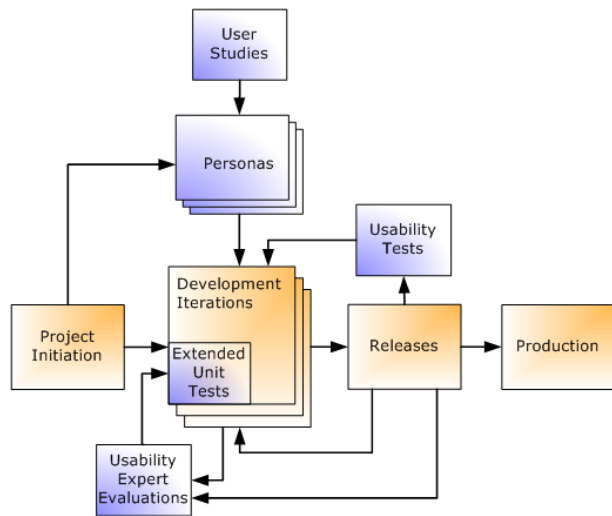


Figure 1. The modified agile development process with user experience instruments included (Extreme Personas, User Studies, Usability Tests, User Experience Expert Evaluations, and Extended Unit Tests). End-users are integrated in two different ways into this process: on the one hand user studies inform the development and extension of the personas (which indirectly provides input to the developers); on the other hand usability tests (as a part of the user experience evaluations) directly inform the development [5][17].

### C. Personas

“Personas are not real people, but they represent them throughout the design process. They are hypothetical archetypes of actual users. Although they are imaginary, they are defined with significant rigor and precision.” [27, pp. 123-124].

Personas are a design tool based on the ideas of Alan Cooper, who released his book “The Inmates are Running the Asylum” in 1999, which is considered to be the founding work in the field of personas [27]. Since the invention of personas, many scientists but also large companies have gathered interest in this approach.

The personas method was developed as a tool for rising empathy for the end users in development teams and as a means for communicating peer group definitions [17][18]. Personas determine what a product should do and how it should behave. They communicate with stakeholders, developers, and other designers. Furthermore, personas build consensus and commitment to the design and measure the design’s effectiveness. They also contribute to other product-related efforts such as marketing and sales plans [27].

They describe the target user – his wishes, desires and application-specific aspects. Furthermore, they show the nature and scope of the design problem [28]. If no personas are defined in a project, the project members will always

envision themselves to be the end-user, which leads amongst others also to communication problems [28]. Each persona represents a peer group of users. With a detailed description of the personas, every member of the project knows the main users and has a unified view of the target customers [27][28].

The advantages of personas are that they allow to unify the picture of the target user for the project team, which allows for a more fluent communication [28]. Additionally, personas make use of the “emotional mind” [29] of people which leads to a better focus on user-centered thinking within a project. “The user” as an abstract term is eliminated from the project and is replaced by people with names and faces, which also saves time because of, e.g., shortened debates. With the personas method, no existing processes have to be modified or changed; personas are just added to the project to focus more on the end-users.

Moreover, personas allow for informed design and according to Alan Cooper, they enlighten the design process [27]. Furthermore, personas can also be used as an evaluation tool such as walkthroughs [28]. As an archetypical figure, personas can guide decisions about product features, navigation, interactions, and even visual design (among other factors) [30].

In the agile development process, personas can be integrated as so-called “Extreme Personas” [17], an approach to personas that starts with the same activities as in the classical persona method: preliminary user groups are defined and personas are modeled according to them. In addition, the knowledge gathered in user studies is incorporated and the personas will be refactored when new knowledge suggests slight changes. If the found knowledge reveals that current personas do not cover the 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 Extreme Personas represent the on-site customer.

### D. User Studies

User studies are the instrument for getting knowledge about end-users. The outcome of the user studies informs the design in two ways: on the one hand knowledge for creating and refactoring the personas is obtained; on the other hand direct input for the user stories can be derived [17][18].

In the agile process, user studies were employed in two different ways [18]:

- Laddering interviews and
- Field trials

The following sections provide a short introduction to both of these areas.

1) *Laddering Interviews*: Laddering interviews are techniques that are mostly applied in the field of marketing and psychology. Nevertheless, recently they have also been applied to investigate user experience [31]. In a structured interview between two persons, the connections between attributes and their consequences are investigated from the interviewee's point of view [32].

The duration of a laddering interview can be – depending on the content – between forty-five minutes and two hours. Structured questions are employed to discover the respondent's beliefs, feelings, and goals. In order to get familiar with the interviewee, a warm-up phase is needed and sessions are usually recorded.

2) *Field Trials*: In field trials the developed products are tested by real users in an uncontrolled setting. The feedback of the users can be collected by applying various techniques. These techniques may include surveys, questionnaires, interviews, contextual inquiries, diary studies, etc.

#### E. Usability Tests

Usability tests are empirical studies that involve real users [18]. They can be considered as the most fundamental user experience evaluation method [33]. Contrary to field trials, usability tests take place in laboratory environments. Usability tests are conducted several times during product development to measure accuracy, user performance, recall-value, and the user's emotional response.

During the tests the users are observed using the product. In some cases the users can be asked to think aloud and verbalize their thoughts to get a better insight to the user's mental model, as well as encountered problems. Other methods such as interviews can be combined with usability tests.

In our project usability tests in the laboratory have included 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 [17].

#### F. Expert-based User experience Evaluations

Expert-based user experience evaluations are reviews conducted by experts [18], either in the area of user experience, in the application area of the particular system, or both. The following two expert-based user experience inspection methods are the most renowned:

- Heuristic Evaluation
- Cognitive Walkthrough

1) *Heuristic Evaluation*: Heuristic Evaluation has gained in popularity with Molich and Nielsen's introduction of ten heuristics to a wider audience [34][35]. The original heuristics have later been improved and adapted to different areas [33] and are still frequently employed to evaluate different kinds of systems.

This evaluation method is considered an efficient analytical and low-cost user experience enhancement method,

which can be applied repeatedly during a development process. In general, heuristics can be considered as rules of thumb describing the affordance of a user to a system and are formulated more generally than the rather specific guidelines. They are recognized and established user experience principles.

During a heuristic evaluation three to five experts (one expert at a time) inspect a system according to given heuristics. The found issues are categorized according to their severity after the evaluation is finished. Heuristics do not cover all possible occurring user experience problems but because of the ease of application they can be employed very early in the design process – even when usability testing would not be possible.

2) *Cognitive Walkthrough*: A cognitive walkthrough is an analytical user experience inspection method which was introduced by Wharton, Rieman, Lewis and Polson [36]. The main goal of a cognitive walkthrough is to measure the learnability of a system by detecting user experience issues. Traditionally this evaluation method is conducted either by a single expert or a group of experts and novice users who put themselves in the place of a hypothetical user [34]. During the evaluation, typical tasks are accomplished within the four phases of a cognitive walkthrough. Similar to heuristic evaluations, cognitive walkthroughs can be applied very easily and early in the design process.

3) *Application in Agile Environments*: In projects involving XP, expert-based user experience 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 mentioned channels [17][18]. For this purpose the methods have to be tweaked in a way to be less time and cost intensive. This is done by involving less experts or users (only 1-2) than recommended. These tests are done much more frequently (every 1-3 iterations) and therefore the results can be accumulated. Important here is to switch users and experts frequently to achieve similar results.

#### G. Extended Unit Tests

Extended unit tests originate from automated usability evaluation (AUE) [18]. The idea of automated usability evaluation is not new. Basic research goes back to the early nineties [37][38]. In the year 2000 the state of AUE was still described as “quite unexplored” [39].

AUE offers usability support through specialized tools. Therefore, developers can be supported by automatic inspection throughout the development phase of a project. An example of such kind of automatic evaluation is log-file analysis [40][41]. Here the generated data helps identify paths and execution time in order to detect problems.

Another example for AUE is the NIST-Web-Suite [50] that allows for an automatic code-based analysis of websites

according to 12 design guidelines. The WAUTR-project [51] (Automatic Usability Testing enviRonment) is a first attempt to support user experience experts with a set of different tools.

The availability of user-generated data already during the development is one problem of AUE. The simulation of the final users [42] or specific aspects (e.g., gaze [52][53]) is one possibility to solve this issue. Nevertheless, also this approach requires user-data.

When no users are available, code-analysis is suggested as the next best solution. The code is used to calculate usability factors and give input on usability based on design guidelines (e.g., the ratio between text and graphics on a website, the number of links, the use of colors, etc.) [39][43].

This approach is currently mainly used in the web area. WebTango is one of the best known tools in this field of application [44]. It calculates usability metrics on basis of HTML code, and the evaluation is based on a statistical model of the website usage.

An approach that goes even further was developed by Abdulkhair [42] who implemented agents that are able to learn from user-behaviors. The underlying statistical model allows the agents to detect user preferences, learn them, and use them to evaluate websites.

The reverse-engineering of the structure of a website was used by Paganelli and Paterno [45] in order to find potential usability problems with their tool “WebRemUSINE”.

Currently, the main target group of such kind of tools are experts in the area of human-computer-interaction. The transition towards developer-based tools for AUE is currently in progress [17]. These tools would then be able to continuously check for user experience issues even during development while the code is being written. Although gaining in popularity, automatic usability evaluation can rarely be found in commercial development environments. The most renowned product of this kind is LIFT [54], which is comparable to WebSAT. Additionally, LIFT integrates the GoLive, FrontPage, and Dreamweaver development environments.

Since most of the current approaches tend to focus on multimodality [46] and mobile devices [47], the existing tools have a big disadvantage for our project: most of them are isolated solutions which are – as mentioned before – solely designed for HCI experts. Hence, they hardly integrate seamlessly into the existing development processes.

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 user experience of the application. Unit tests – by definition – test small definable units of the software. The problem of patchwork application suggests using a holistic approach to testing. Therefore, unit tests are extended by tests that go beyond single units and test complete interaction flows [17][18].

## V. CONCLUSIONS AND LESSONS LEARNED

We used the previously described process since summer 2007 in our project which ended in summer 2010. The final usability tests, diary studies, focus groups, and the results of log file analysis in the field trial held from December 2008 till May 2009 show that the process is able to really enhance user experience of XP-style developed applications [48].

The tight coupling of different expertise has led to high motivation among project members. Developers gain insight into the subtleties of UCD and HCI experts learn to understand the origin of user experience problems. Especially the diverse technical testing frameworks demand technically aware HCI experts. In practice, this could become a problem when the chosen framework is complex and little time for learning is available [5][17][18].

We have experienced that the inclusion of UCD in the software development processes underpins past experiences: no matter if it is a classical “waterfall” development process or XP – the inclusion of UCD mainly depends on the user experience-awareness and on the mindset present in the project – not on the software development model.

Furthermore, we found that especially ad hoc input can be given sufficiently via mail since most of the time no synchronous communication between the project members is needed. The geographical distance between HCI practitioners and developers can partly be overcome by using phone- or video-conferencing [18].

Also from a customer point of view the communication with a user experience engineer can be done most of the time by e-mail and exchanging mock-ups. Especially for mock-ups it turned out to be important that the user experience engineer actually sees them rather than getting them described. The response times of these methods are usually short enough since for user experience input in the story-writing process it is sufficient to get results within 3 to 4 days. For quick feedback on user experience issues during development or for “urgent” re-planning, user experience engineers should be readily available for a quick advice via cell-phone or chat [18]. However we also had good experiences with on-site visits. This allows the user experience engineer to see in which environment the application is developed and to get a better understanding of the developers and customers.

In our case, the creation of user-stories is supported by HCI knowledge derived from studies, literature, and usability

tests. During our project the story-writing was done mainly by the customer and the technicians prior to planning.

A technician of our team paired with the customer in order to create simple and precise stories for the discussion in the planning game. This pairing forces the customer to explain his expectations in detail. The technician helps to refine the ideas by asking questions which could determine implementation details and gives feedback on the answers. Together they ensure that the story is written in a way which is unambiguous and understood by both sides. This way a lot of time and energy, which would have been spent during team discussions, is saved during the planning game. If the story contains user interface (UI) aspects, the user experience engineer is also included early into the story creation-process. The advantages are timesaving, more motivation (less chances of rework due to preclarified user experience issues), and gaining better understanding of needed usability input early in the development [18].

Continuous monitoring, evaluating, and testing of the UI, and quick intervention can lower the danger of a patchwork-experience. Additionally, we could see that cultural problems between HCI and development seem to depend more on the involved persons than the methods used, and we did not experience the problems reported in the literature.

#### A. Review of the Used Instruments

During this project we have extended traditional XP methods with knowledge derived from different HCI instruments. In the following sections, we provide a review of some the methods employed:

1) *Extreme Personas*: The personas method should enable an end-user focused mindset to be established very quickly and hence should solve the problem of the development focus on the technical part. Additionally, the personas should help to orient the project towards on-site customer AND end-user [18]. During our project we concentrated very much on the customer centered design process as well as the design process itself and nearly neglected the personas.

There have been several issues and discussions on Extreme Personas during our project. First, the initially developed personas were not satisfactorily distributed to either the development team or the customer-on-site by the user experience engineers. Second, the development team did not give much credit to the two personas which were provided [18].

From the point-of-view of the development team and the customer-on-site the main cause for these issues was that especially one persona was so funny that they did not take it serious – nevertheless, they got in touch with them instead of neglecting them fully [18].

*What can be learned out of this?* First, Personas should be introduced like new team members, as they will accompany the whole team during the development process. Therefore, they need “room” and “positive energy”: they should give the

developers and customers a feeling of producing something valuable for someone who they like. Of course, fun can (and should) play a major role within teams – but be careful not to mob a persona!

Second, many technicians think that it does not matter for whom they develop. The design is supposed to be the design-department’s decision, the scope of implemented features is part of the customer’s work, and over-all everything is pre- and post-tested by the user experience engineer anyhow – so why worry? If this would be true, why did all technicians in our development team take part in a months-lasting discussion about the user-group we were developing and producing for? Especially the customers thought that the technical or business-related decision processes are colored by conscious and unconscious inputs. That is why personas should be present in an appropriate form and should have their own stable place around the developers. Stories and features should be developed for the personas and their names should be used on the story cards and during discussions. This will help them stay alive and influence the team.

Therefore, the advice from the customer-side is to make personas available and visible, take part in the process of developing them, introduce them to the team, and have them (consciously!) in mind when planning or undertaking any decision process.

2) *Expert Evaluations*: During our project, it took some time until the customers embraced the possibility of asking the advice of the user experience engineer in advance. This might have been caused by the increasing trust in the user experience engineer over time. After a while the customers have reportedly valued the user experience engineer’s input higher. Another reason was the improved planning method which allowed to prepare stories early enough to have time for user experience input.

The experience with expert evaluation was a very positive one if the results arrived on time. For an XP project the usual way expert evaluations are done is not ideal. Instead of big, long lasting, application wide evaluations, what is needed are smaller and faster evaluations on the story level. If the evaluation result comes after the story or iteration completion, the likelihood of ignoring the input increases dramatically. The reason is that either the input is already outdated due to the quick changes in an agile project, or other stories are higher prioritized by the customer. Consequently, a stripped down, quicker version of the usual expert evaluation process is needed.

*What can be learned out of this?* First: Prepare the user stories at least three days before the actual planning. During this preparation when the story-cards are written, all the mock-ups should be drawn as well (use drawings by hand or simple drawing tools). Then the mock-ups are sent to the user experience engineer for feedback [18].

Feedback from the user experience-side should be quick (maximum 3 days for one week iterations). The advantage:

when introducing the story – it is already usability tested. When technical questions arise during the implementation, for instance that a certain demand from the user experience side would cost too much, it is advised to call the user experience-engineer and have a short (video) conference during planning or whenever this occurs. Far less user experience-fixes are the results of this practice [18].

Thus, it is our advice to involve available user experience engineers into the planning process as early as possible. Use his time and input only when the implementation of the results is immanent or when it is critical for the development. Be careful: Do not shift the evaluation and fixes to “when you have time” because this will never occur and thus no serious user experience input will be realized.

3) *Usability Tests:* We noticed that the usability tests had impact beyond the expected one (which is giving input for the design). We saw that the mindset of developers changed dramatically when seeing real users handling the application during these tests [5][18]. We saw that developers who attended the usability tests got more biased towards user-centered thinking than the others [5]. Some of the developers not having been present wanted to watch the recordings of the tests but did not find the time to do so.

When it comes to the results of the tests there was an agreement that the tests were too early in the project to tell us a lot about the user experience problems of the application. Furthermore, the reporting period was too long. When the report arrived there already had been so many changes in the application that many recommendations were obsolete [18].

Therefore, we would recommend smaller tests after every few iterations of the application (better: for every iteration). To keep costs low and if the system is a very fast moving target, not always the entire system should be tested and the number of test users can also be limited to 1-2. This is compensated by the increased frequency of tests resulting in a similar coverage than a big test. Bigger tests should only be made when no major changes are expected and the system is quite stable [18].

4) *User Studies:* We used user studies in the form of laddering interviews and field studies. In autumn 2007 laddering interviews were conducted. From December 2008 till May 2009, a large field trial study was conducted with 150 real end-users spread throughout Austria who used our application on mobile phones.

The users were able to use their devices freely to access multimedia content and did not have any restrictions. They were only asked to fill in questionnaires sent to their mobile phones and reply to certain SMS. The trial study also included diary studies, contextual interviews, laboratory usability tests, and focus groups. [18]. We also logged the actions of the users. The preliminary log files results were presented in [48].

5) *Lightweight Prototypes:* Besides the HCI instruments mentioned above, we also made use of two different types of mock-ups: Low fidelity paper mock-ups, and high fidelity mock-ups if a more detailed clarification was necessary. We got them evaluated by the customer and afterwards sent them to the user experience engineer for additional feedback [18].

## B. Tips & Tricks for Practitioners

Since we have gathered some experience in the combination of XP and HCI centered methods we are providing some tips and tricks in this area.

From the customer’s point of view it is recommended to learn to cope with possible daily occurring misunderstandings between business and technical team members. During the planning process clarification is particularly important.

Furthermore, the user experience expertise should be integrated very early in the planning phase and also during the story-writing process. Invite your user experience engineer from time to time or visit him during user testing. It is quite a lesson watching real users acting with your prototype the first time and giving feedback.

1) *Tips & Tricks for Waterfall-model Projects:* In waterfall-model projects we propose to follow these basic guidelines for including UCD in the development process:

- End-user inclusion: It is important to include end-users as early as possible. It is best to start cooperating already during the requirement phase.
- Pre-requirements phase: as soon as targeted user-groups are known an HCI engineer should do research on existing knowledge of the interaction behavior.
- Prototyping: Use paper-prototyping to gather early feedback from end users. Tests of the paper mock-ups of GUI designs prior to the implementation are particularly important in waterfall-model projects.
- Team building: Include not only end-users but also HCI experts from the beginning. Requirements should be backed by HCI input to ensure that architectural design of a system does not include user experience threats from the ground-up. Some architectures forbid to hide unnecessary complexity from the user which is why HCI input is also important for technology-centered architectural decisions.

2) *Tips & Tricks for Extreme Programming Projects:* Boiled down to a short list we summarize the actual state of experience:

- Usability Test Videos: A highlight video of the test should be created. Highlight videos save time compared to the full video documentation of usability tests and support a user-centered mind-set. The highlight video should be shown to the whole development team.
- Training: HCI engineers should be trained in XP-story writing to be able to deliver their user experience findings in form of user-stories. This saves an additional

step and makes them immediately usable during the planning game [18].

- Story writing: a developer should support the HCI engineer when writing user-stories [18].
- Proper customer and HCI engineer coordination is necessary for the inclusion of the user experience process in the development [18].
- An experienced on-site XP customer can fill-in the technical gap between HCI engineers and developers.
- A short user experience workshop should be held at the start of the project or before the release in which the usability testing phase begins. It serves as a good platform for all XP team members (managers, customers, and developers) to understand the process and importance of usability testing. It also serves as a good kiff-off time for HCI engineers and developers as they have a lot of work to do together in future.
- A pre-plan should be developed by the HCI engineer that states at which detail level of UI design development to use low-fidelity and high-fidelity mock-ups and when to perform which usability testing process. This will help the customer in planning for the iterations. A short meeting of the HCI engineer and the customer before an iteration planning will further help the customer in UI design stories.
- It is the customer's responsibility to make the up-to-date usability tests reports visible to the developers (either on a dedicated usability-tests board or somewhere near the story board).
- As the customer is the one creating and prioritizing the stories it is his duty to also think about and include user experience aspects in his user stories.

### C. Summary & Future Research Directions

We have been able to show that different HCI methods and techniques impact the user experience of XP projects as intended. As we saw, all need tweaking and fine-tuning to perfectly fit the XP process.

For the future we see different directions of research. One is automated usability evaluation (AUE) and its integration into software development processes. Research on testing-frameworks for AUE will be an important next step where HCI experts and developers should collaborate to ensure the final frameworks fulfill requirements of both disciplines. The need for better AUE is obvious: the quickly growing number of custom software products is not matched by a similar growing number of HCI experts. Hence, tools will be needed to support HCI engineers in handling these. As purely code-based AUE approaches are limited to a certain level, more advanced AUE methods have to be developed.

A second research direction for us concerns the need for more in-depth insight into the persona-method. We have to gain more knowledge about the interrelationship between the modeled personas and the cognitive effects on different

developers. One issue is the perception of persona pictures: theory suggests personas should be "likable" – but what do developers like and dislike? Hence, research in the perception of personas should be broadened. Open questions for us are for example: which features of modeled personas support which outcome in the development? What about the influence of different subjective perceptions of a persona? Collaborating with different disciplines – e.g., the game industry, as they know a lot about "character modeling" or media scientists who model characters for TV series – will be necessary to cover these questions.

On the business side we see the need for a more elaborated process on how to include different stakeholders and their input. We assume that the more stakeholders get involved the higher the need for structured inclusion strategies for all stakeholders will become. Research on these inclusion strategies will be necessary to ensure that the input of each stakeholder is treated the right way.

### ACKNOWLEDGMENT

The research herein was partially conducted within the competence network Softnet Austria ([www.soft-net.at](http://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

- [1] M. Lechner, "Curid – a software development method for data-driven framework architectures," Master's thesis, Technical University of Graz, Austria., 2005.
- [2] C. Larman and V. R. Basili, "Iterative and incremental development: A brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003.
- [3] K. Beck, *Extreme Programming Explained: Embrace Change (1st Edition)*. Addison-Wesley Professional, 1999.
- [4] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change (2nd Edition)*. Boston: Addison-Wesley, 2004.
- [5] Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, and P. Wolkerstorfer, "Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application," in *USAB 2008*, ser. LNCS, vol. 5298/2008. Springer Berlin / Heidelberg, November 2008, pp. 313–330.
- [6] W. Hudson, "A tale of two tutorials: a cognitive approach to interactive system design and interaction design meets agility," *interactions*, vol. 12, no. 1, pp. 49–51, 2005.
- [7] P. McInerney and F. Maurer, "UCD in agile projects: dream team or odd couple?" *Interactions*, vol. 12, no. 6, pp. 19–23, 2005.



- [8] J. Patton, "Hitting the target: adding interaction design to agile software development," in *OOPSLA 2002 Practitioners Reports*. Seattle, Washington: ACM, 2002.
- [9] S. Chamberlain, H. Sharp, and N. Maiden, "Towards a framework for integrating agile development and user-centred design," in *7th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP 2006*, ser. LNCS, vol. 4044. Heidelberg, Germany: Springer Verlag, 2006, pp. 143–153.
- [10] A. Holzinger, M. Errath, G. Searle, B. Thurnher, and W. Slany, "From extreme programming and usability engineering to extreme usability in software engineering education (XP+UE→XU)," in *COMPSAC '05: Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05) Volume 2*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 169–172.
- [11] A. Holzinger and W. Slany, "(XP+UE→XU) praktische erfahrungen mit extreme usability," *Informatik Spektrum*, vol. 29, no. 2, pp. 91–97, 2006.
- [12] D. Sy, "Adapting usability investigations for agile user-centered design," *Journal of Usability Studies*, vol. 2, no. 3, p. 112132, 2007.
- [13] J. Ferreira, J. Noble, and R. Biddle, "Agile development iterations and UI design," in *Agile 2007*. IEEE Computer Society, 2007, pp. 50–58.
- [14] J. Ferreira, J. Noble, and R. Biddle, "Up-front interaction design in agile development," in *8th International Conference on Agile Processes in Software Engineering and eXtreme Programming, XP 2007, Jun 18-22 2007*, vol. 4536 LNCS. Heidelberg, D-69121, Germany: Springer Verlag, 2007, pp. 9–16.
- [15] D. Fox, J. Sillito, and F. Maurer, "Agile methods and User-Centered design: How these two methodologies are being successfully integrated in industry," in *Agile, 2008. AGILE '08. Conference, 2008*, pp. 63–72.
- [16] S. W. Ambler, *Tailoring Usability into Agile Software Development Projects*. Springer London, 2008.
- [17] P. Wolkerstorfer, M. Tscheligi, R. Sefelin, H. Milchrahm, Z. Hussain, M. Lechner, and S. Shahzad, "Probing an agile usability process," in *CHI '08: human factors in computing systems*. New York, USA: ACM, 2008, pp. 2151–2158.
- [18] Z. Hussain, H. Milchrahm, S. Shahzad, W. Slany, M. Tscheligi, and P. Wolkerstorfer, "Integration of extreme programming and user-centered design: Lessons learned," in *XP 2009*, ser. LNBIP, P. Abrahamsson, M. Marchesi, and F. Maurer, Eds., vol. 31. Springer, 2009, pp. 174–179.
- [19] Z. Hussain, W. Slany, and A. Holzinger, "Current State of Agile User-Centered Design: A Survey," in *USAB 2009*, LNCS, vol. 5889. Springer, 2009, pp. 416–427.
- [20] L. Miller and D. Sy, "Agile user experience SIG," in *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems CHI*. Boston, MA, USA: ACM, 2009, pp. 2751–2754.
- [21] M. Budwig, S. Jeong, and K. Kelkar, "When user experience met agile: a case study," in *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems CHI*. ACM, 2009, pp. 3075–3084.
- [22] J. T. Barksdale and D. S. McCrickard, "Concept mapping in agile usability: a case study," in *CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. ACM, 2010, pp. 4691–4694.
- [23] J. Ferreira, H. Sharp, and H. Robinson, "Values and assumptions shaping agile development and user experience design in practice," in *XP 2010*, ser. Lecture Notes in Business Information Processing, vol. 48. Springer, 2010, pp. 178–183.
- [24] K. Tzanidou and J. Ferreira, "Design and development in the "agile room": Trialing scrum at a digital agency," in *XP 2010*, ser. Lecture Notes in Business Information Processing, vol. 48. Springer, 2010, pp. 372–378.
- [25] J. Ding, C. Lin, and K. Huang, "ARS: An adaptive reception scheme for handheld devices supporting mobile video streaming services," in *International Conference on Consumer Electronics. ICCE '06*, vol. 1, 2006, pp. 141–142.
- [26] Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, T. Vlk, and P. Wolkerstorfer, "User interface design for a mobile multimedia application: An iterative approach," in *ACHI 2008, First International Conference on Advances in Computer-Human Interaction, Sainte Luce, Martinique, France*. IEEE Computer Society, February 10–15, 2008, pp. 189–194.
- [27] A. Cooper, *The Inmates Are Running the Asylum*. Indianapolis, IN, USA: Macmillan Publishing Co., Inc., 1999.
- [28] J. Pruitt and T. Adlin, *The Persona Lifecycle: Keeping People in Mind Throughout Product Design (The Morgan Kaufmann Series in Interactive Technologies)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2006.
- [29] L. Shyba and J. Tam, "Developing character personas and scenarios: vital steps in theatrical performance and hci goal-directed design," in *C&C '05: Proceedings of the 5th conference on Creativity & cognition*. New York, NY, USA: ACM, 2005, pp. 187–194.
- [30] J. E. Nieters, S. Ivaturi, and I. Ahmed, "Making personas memorable," in *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2007, pp. 1817–1824.
- [31] M. Pifarré and O. Tomico, "Bipolar laddering (bla): a participatory subjective exploration method on user experience," in *DUX '07: Proceedings of the 2007 conference on Designability for User eXperiences*. New York, NY, USA: ACM, 2007, pp. 2–13.
- [32] T. J. Reynolds and J. C. Olson, Eds., *Understanding Consumer Decision Making: The Means-end Approach to Marketing and Advertising Strategy*. Mahwah, NJ: Lawrence Erlbaum, 2001.

- [33] J. Nielsen, *Usability Engineering*. Morgan Kaufmann, 1993.
- [34] T. Hollingsed and D. G. Novick, "Usability inspection methods after 15 years of research and practice," in *SIGDOC '07: Proceedings of the 25th annual ACM international conference on Design of communication*. New York, NY, USA: ACM, 2007, pp. 249–255.
- [35] R. Molich and J. Nielsen, "Improving a human-computer dialogue," *Commun. ACM*, vol. 33, no. 3, pp. 338–348, 1990.
- [36] C. Wharton, J. Rieman, C. Lewis, and P. Polson, "The cognitive walkthrough method: a practitioner's guide," pp. 105–140, 1994.
- [37] S. Balbo, J. Coutaz, and D. Salber, "Towards Automatic Evaluation of Multimodal User Interfaces," in *IUI '93: Proceedings of the 1st International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM Press, 1993, pp. 201–208.
- [38] M. L. Hammontree, J. J. Hendrickson, and B. W. Hensley, "Integrated data capture and analysis tools for research and testing on graphical user interfaces," in *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 1992, pp. 431–432.
- [39] M. Y. Ivory and M. A. Hearst, "The state of the art in automated usability evaluation of user interfaces," EECS Department, University of California, Berkeley, Tech. Rep. UCB/CSD-00-1105, Jun 2000. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2000/5557.html>
- [40] R. Atterer and A. Schmidt, "Tracking the interaction of users with ajax applications for usability testing," in *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM, 2007, pp. 1347–1350.
- [41] J. I. Hong and J. A. Landay, "WebQuilt: A Framework for Capturing and Visualizing the Web Experience," in *WWW '01: Proceedings of the 10th International Conference on World Wide Web*. New York, NY, USA: ACM Press, 2001, pp. 717–724.
- [42] M. Abdulkhair, "A multilingual automated web usability evaluation agent," Ph.D. dissertation, University of Sheffield, England, 2004.
- [43] M. Y. Ivory and M. A. Hearst, "Improving web site design," *IEEE Internet Computing*, vol. 6, no. 2, pp. 56–63, 2002.
- [44] M. Y. Ivory, "Web TANGO: Towards Automated Comparison of Information-centric Web Site Designs," in *CHI '00: CHI '00 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2000, pp. 329–330.
- [45] L. Paganelli and F. Paterno, "Automatic reconstruction of the underlying interaction design of web applications," in *SEKE '02: Proceedings of the 14th international conference on Software engineering and knowledge engineering*. New York, NY, USA: ACM, 2002, pp. 439–445.
- [46] F. Patern, A. Piruzza, and C. Santoro, "Remote web usability evaluation exploiting multimodal information on user behavior," in *CADUI*, G. Calvary, C. Pribeanu, G. Santucci, and J. Vanderdonckt, Eds. Springer, 2006, pp. 287–298.
- [47] S. J. Waterson, J. I. Hong, T. Sohn, J. A. Landay, J. Heer, and T. Matthews, "What Did They Do? Understanding Click-streams with the WebQuilt Visualization System," in *AVI '02: Proceedings of the Working Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2002, pp. 94–102.
- [48] Z. Hussain and W. Slany, "Analyzing real mobile web usage of a multimedia streaming application through log files," in *MIR '10: Proceedings of the international conference on Multimedia information retrieval*. ACM, 2010, pp. 401–404.
- [49] <http://www.agilemanifesto.org/> (last accessed 2011/12/01).
- [50] <http://zing.ncsl.nist.gov/WebTools/> (last accessed 2011/12/01).
- [51] <http://wauter.weeweb.com.au/> (last accessed 2011/12/01).
- [52] <http://www.goodgaze.com/ggx/> (last accessed 2011/12/01).
- [53] <http://www.feng-gui.com/> (last accessed 2011/12/01).
- [54] <http://www.usablenet.com/> (last accessed 2011/12/01).