Zahid Hussain

Empirical Analysis of Agile User-Centered Design

Dissertation

vorgelegt an der

Technischen Universität Graz, Austria

zur Erlangung des akademischen Grades
Doktor der Technischen Wissenschaften
(Dr.techn.)

durchgeführt am Institut für Softwaretechnologie
Technische Universität Graz, Austria

Betreuer & Begutachter 1: Univ.-Prof.Dr.techn.Dipl.-Ing. Wolfgang SLANY
Begutachter 2: Univ.Doz. Ing. Mag.rer.nat. Mag.phil. Andreas HOLZINGER

Graz, im Juli 2010

Zahid Hussain

Empirical Analysis of Agile User-Centered Design

Dissertation

submitted to the

Graz University of Technology, Austria

for the partial fulfillment of the requirements for the academic degree of
Doctor of Philosophy
(Ph.D)

at the Institute for Software Technology
Graz University of Technology, Austria

Supervisor & Assessor 1: Univ.-Prof.Dr.techn.Dipl.-Ing. Wolfgang SLANY
Assessor 2: Univ.Doz. Ing. Mag.rer.nat. Mag.phil. Andreas HOLZINGER

Graz, July 2010

"To my Family whose love and passion always encouraged me!"

Acknowledgments

Praise be to Almighty God (Allah), the Beneficent, the Merciful, who has awarded me with this great achievement of my life.

I dedicate this achievement to my parents who inspired me and helped me in my studies from my childhood. I also dedicate and share my joy with my family, brothers, and sisters. I find no words of thanks for my family: my wife and my children, and my brother's family and his children.

I owe my deepest gratitude to my supervisor, Professor Dr. Wolfgang Slany, for his constant encouragement, help, and invaluable supervision of my research. His encouragement in different aspects of the work contributed to my personal and professional development. Lots of sincere wishes and thanks are due to my external supervisor, Professor Dr. Andreas Holzinger, for his guidance, support, and encouragement in the final and critical stages of my research. Many thanks to my colleagues at the Institute for Software Technology whose help, support and guidance has a big share in the success of my PhD studies, particularly Sara Shahzad, Martin Lechner, Martin Umgeher, Harald Milchrahm, and Thomas Vlk. Many thanks also to our institute's secretary Petra Pichler and our technician Herbert Pöckl for their help and for their nice company. My gratitude also goes out to my Pakistani colleagues in Graz, Austria especially to Muhammad Saleem Awan, Syed Nadeem Ahsan, Hasnat Ahmad Malik, Shahzad Saleem, Mudassar Abbas, Syed Tanweer Iqbal, and Tajamal Hussain who have helped me through discussions, guidance and providing their nice company.

My heartiest tribute to my country Pakistan and Higher Education Commission (HEC), Government of Pakistan, for partially funding my research, and to the Austrian Agency for International Cooperation in Education and Research (ÖAD-GmbH). Special thanks to my employer, Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah, Pakistan, on their positive role during my PhD studies and to the Institute of Software Technology, Graz University of Technology for providing all possible support required for my studies. It is my aim to make a positive contribution and to play an active role in the prosperity and fame of my country. During the thesis, a lot of people have been very supportive and therefore are due my biggest thanks. Thanks to anybody I missed who deserves a mention! May God give reward to all those who helped me and prayed for me during my research. Aameen.

Zahid Hussain

Abstract

This thesis investigates the relationship between agile software development methods and user-centered design. Agile methods are being adopted at an increasing rate in industry every year. However, there is evidence that shows that these methods lack usability awareness in their development lifecycle. So focus on the integration of these two important disciplines in software development is increasing. The main research question of this thesis was: How successfully are these important methodologies being integrated? This research question was further divided into sub-questions: How the role of a user-centered design professional is carried out in organizations? What types of HCI techniques are used in teams/projects employing agile methods? To what degree are agile development methods and user-centered design perceived by practitioners as being integrated and added value to their adopted processes and teams? The research methods that were used for this thesis are: an online survey, interviews from several agile teams and analyzed through Grounded Theory, and a case study of the project where several HCI techniques were introduced in Extreme Programming agile process for more than two years. The multimedia application that was developed in the project enables the user to search in video content and plays the video on mobile phones.

The findings show that the majority of the practitioners have dedicated user-centered design professionals co-located in their teams. If not present then, this role is performed by developers having an interest in HCI. The use of low-fidelity prototyping, usability expert evaluations, and rapid iterative testing easily fit within the fast moving iterations of agile methods. The top most used HCI techniques are low-fidelity prototyping, conceptual designs, observational studies of users, usability expert evaluations, field studies, personas, rapid iterative testing, and laboratory usability testing. The practitioners perceive that the integration of agile methods with user-centered design has added value to their adopted processes and to their teams, has resulted in the improvement of usability and quality of the products developed, and has increased the satisfaction of end-users of the products developed. This is also proved in the field trial and usability test results of the application that show that the overall user experience of the developed application has significantly improved. The findings give hope and new prospective for better usability and quality of the products developed through this integrated methodology.

Deutsche Kurzfassung der Dissertation

Die vorliegende Arbeit untersucht die Beziehung zwischen agilen Softwareentwicklungsmethoden und User Centered Design. Agile Methoden werden in zunehmendem Maße in der Industrie angewandt. Es gibt jedoch Hinweise darauf, dass diese Methoden mangelndes Bewusstsein in Hinblick auf Benutzerfreundlichkeit zeigen. Daher werden diese beiden wichtigen Disziplinen in der Softwareentwicklung miteinander in zunehmenden Maße kombiniert und integriert. Die zentrale Forschungsfrage dieser Arbeit war: Wie erfolgreich werden diese beiden wichtigen Methoden integriert? Die Fragestellung wurde weiter in Unterfragen unterteilt: Wie wird die Rolle eines professionellen User Centered Design Experten in Softwareentwicklungsprojekten ausgefüllt? Welche Arten von Usability Techniken werden in Projekten mit agilen Methoden kombiniert? Bis zu welchem Grad werden agile Entwicklungsmethoden und Usability von Praktikern als integriert wahrgenommen und bilden einen Mehrwert für ihre Prozesse? Die Forschungs-Methoden, die für diese Arbeit verwendet wurden, sind: eine Online-Umfrage, Interviews aus mehreren agilen Teams, analysiert durch Grounded Theory, und eine Fallstudie des Projekts, bei dem mehrere HCI Techniken in dem agilen Prozess Extreme Programming über mehr als zwei Jahren hindurch eingeführt wurden. Die Multimedia-Anwendung, die im Rahmen des Projekts entwickelt wurde, ermöglicht dem Benutzer die Suche in Video-Content und spielt das Video auf dem Handy ab.

Die Ergebnisse zeigen, dass die Mehrheit der Praktiker dedizierte Spezialisten im Team haben, die vor Ort mit ihren Teams zusammenarbeiten. Die Verwendung von Low-Fidelity Prototyping, einfaches und hufiges Feedback durch Usability-Experten und schnelle iterative Tests passen leicht zu den kurzen Iterationen der agilen Methoden. Die meisten HCI Techniken die eingesetzt werden sind Low-Fidelity Prototyping, konzeptionelle Entwrfe, Beobachtungsstudien von Benutzern, Usability-Experten Auswertungen, Feldstudien, Personas, schnelles iteratives Testen und Usability Tests im Labor. Die Praxis lsst erkennen, dass die Integration von agilen Methoden mit User Centered Design einen Mehrwert bedeutet, in der Verbesserung der Benutzerfreundlichkeit und Qualität der entwickelten Produkte mndet, und die Zufriedenheit der End-Anwender mit den Produkten zugenommen hat. Die Ergebnisse geben Hoffnung auf eine bessere Qualität der Produkte.

Contents

Abstract **1**

Kurzfassung (abstract in German) **2**

Table of Contents **3**

List of Tables **6**

List of Figures **8**

1 Introduction **9**
1.1 Agile Software Development Methods 9
1.1.1 Extreme Programming 10
1.2 User-Centered Design 12
1.3 Motivation for Integrated Agile User-Centered Design Process . 14
1.3.1 Why do some XP projects fail when including Human-Computer Interaction (HCI) work? 14
1.3.2 Similarities between XP and UCD 15
1.4 Problem Statement 17
1.5 Major Results of this Thesis 17
1.6 List of Publications 18
1.7 Structure of this Thesis 19

2 Method **21**
2.1 Introduction 21
2.2 Survey 21
2.3 Interviews using Grounded Theory 22
2.4 Case Study 23
2.4.1 The Application 23

3 Agile Software Development Methods and User-Centered Design: Perspectives from a Survey 28
3.1 Introduction . . . 29
3.2 Related Literature Studies . . . 30
3.2.1 Related Studies on Agile Usability/UCD . . . 30
3.2.2 Related Studies regarding Surveys on Agile Methods and Usability/UCD . . . 32
3.2.3 Research Questions . . . 33
3.3 Method . . . 34
3.4 Results and Discussion . . . 35
3.4.1 RQ1: Role of UCD Professional . . . 42
3.4.2 RQ2: HCI Techniques Used: . . . 43
3.4.3 RQ3: The Impact of the Integration of Agile Methods and Usability/UCD: . . . 44
3.4.4 Limitations . . . 46
3.5 Conclusion . . . 47

4 Investigating Agile User-Centered Design in Practice: A Grounded Theory Perspective 49
4.1 Introduction . . . 49
4.2 Related Literature Studies . . . 50
4.3 Method . . . 52
4.3.1 Participants . . . 54
4.4 Results . . . 56
4.5 Conclusion . . . 59

5 Integration of Extreme Programming and Usability / User-Centered Design: A Case Study 60
5.1 Case Study . . . 60
5.1.1 Approach . . . 61
5.1.2 Personas . . . 64
5.1.3 User Studies . . . 66
5.1.4 Usability Tests . . . 66
5.1.5 Expert-based Usability Evaluations . . . 67
5.1.6 Extended Unit Tests . . . 68
5.2 HCI Instruments: Lessons Learned . . . 70
5.2.1 Review of the Used Instruments . . . 71
5.2.2 Extreme Personas . . . 72

5.2.3 Expert Evaluations . . . 73
5.2.4 Usability Tests . . . 74
5.2.5 User Studies . . . 80
5.2.6 Extended Unit Tests . . . 81
5.2.7 Lightweight Prototypes . . . 81
5.2.8 Tips & Tricks for Practitioners . . . 82
5.3 XP Practices: Lessons Learned . . . 84
5.3.1 Testing Issues . . . 84
5.3.2 Communication and Collaboration . . . 84
5.3.3 The Planning Game . . . 85
5.3.4 Pair programming . . . 86
5.3.5 On-site Customer . . . 87
5.3.6 Small Releases . . . 87
5.3.7 Sit Together . . . 87
5.3.8 Collective Ownership . . . 88
5.3.9 40-hour Week . . . 88
5.3.10 Test-first Programming . . . 88
5.3.11 Reflection . . . 89
5.4 Summary & Future Research Directions . . . 89

6 Analyzing Real Mobile Web Usage of a Multimedia Streaming Application Through Log Files 91
6.1 Introduction . . . 92
6.2 The Field Trial Study . . . 93
6.2.1 Selection of Users . . . 93
6.2.2 Study Results . . . 94
6.3 Conclusion . . . 94

7 Conclusion and Future Research Directions 96
7.1 General Conclusions . . . 96
7.1.1 The mobile multimedia streaming application . . . 97
7.1.2 Future Research Directions . . . 97

Bibliography 99

Appendix 115

List of Tables

3.1 The various job titles of the 92 respondents 35
3.2 Locations of the 92 respondents 35
3.3 Total experience of the 92 respondents in software development 36
3.4 Experience of the 92 respondents in agile methods 36
3.5 Industry sector of the respondents 36
3.6 Total number of employees working in the organizations of the respondents 37
3.7 Total number of members working in the teams of the respondents 37
3.8 Some of the agile practices adopted by and some of the roles carried out in the teams of the respondents 38
3.9 The various agile methods used (multiple answers possible) . . . 39
3.10 Ratings of the Face-2-Face communication channel for conveying information with a UCD professional 40
3.11 Ratings of the Teleconference Calls communication channel for conveying information with a UCD professional 40
3.12 Ratings of the Online Chat communication channel for conveying information with a UCD professional 40
3.13 Ratings of the Video Conferencing communication channel for conveying information with a UCD professional 41
3.14 Ratings of the Email communication channel for conveying information with a UCD professional 41
3.15 when HCI / usability evaluations occur in projects (multiple answers possible) 41
3.16 Comparison between the requirement details whether they emerge over time on projects or whether they are fixed up-front 41
3.17 If the requirement details emerge over time then whether some UI design up-front is done before any development begins in projects 42

3.18 The role of a UCD professional carried out in the organizations of the respondents 42
3.19 Comparison of the approach of co-located versus remotely located UCD professional 43
3.20 Comparison of the approach of having dedicated UCD professional versus this role performed by a developer having interest in HCI/usability 43
3.21 Whether it is possible that a developer can pick-up HCI skills . 44
3.22 How can a developer learn HCI skills? (multiple answers possible) 44
3.23 The various HCI Techniques used (multiple answers possible) . . 45
3.24 The integration of agile methods with usability/UCD has added value to the adopted process and to the teams 45
3.25 The adoption of an agile UCD process has resulted in the improvement of usability and quality of the product developed . . 46
3.26 The resulting product has increased the satisfaction of its end-users due to the adoption of an agile UCD process 46

5.1 Subjective metric (Shodan 2.0 Input Metric Survey) 90

List of Figures

2.1 Home page of the application. 24
2.2 Search results page showing the context of the spoken words found in the clip and showing their position with graphical markers on a timeline of the clip. 26

5.1 The modified agile development process with usability instruments included (Extreme Personas, User Studies, Usability Tests, Usability 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 usability evaluations) directly inform the development [145][68]. 62
5.2 Iterative UI design workflow [65]. 63
5.3 The prototype of the home page 75
5.4 The prototype of the Channel page showing the calender 76
5.5 The menu entries without any visual separation 77
5.6 Improvements of menu layout and arrangement 79
5.7 Use the space on top of the “Clip Detail” more efficiently . . . 80
5.8 Selected story cards on the Release-Board (Release Planning) . 86
5.9 Selected story cards on the Iteration-Board (Iteration Planning) 86
5.10 Executable code versus test code and coverage 88

Chapter 1

Introduction

This chapter provides an introduction to agile software development methods and User-Centered Design (UCD). Then the motivation for this research work regarding the integration of Agile UCD/Usability methodologies is described which is followed by the problem statement. The list of the publications that resulted during carrying out this research work is also given. Finally, the structure of this thesis is described.

1.1 Agile Software Development Methods

The number of software projects constantly increases but the overall success rate is still rather low [92]. Many projects fail because of their inability to cope with the changing user requirements. Heavy up-front design without continuous feedback from the customer is another factor for this failure. The intention of large scale research into software engineering techniques has been a formulation of an ideal methodology that can consistently and predictably lead to software development success [96]. To have a greater probability of success, the developers need a software development process which should be flexible enough to cope with the constantly changing requirements and which is also people-oriented. Agile software development methods have emerged in response to these needs as agile methods give more value to individuals, working software and change [91]. In software engineering, 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. 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. The Agile Manifesto [91] signed by renowned software methodologists in 2001 describes four values and twelve principles. The four values are:

"**Individuals and interactions** over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan"

While recognizing the importance of the items on the right, the emphasis is on the items of the left side [91]. Ambler [9] gives a comprehensive definition of disciplined agile software development:

"An iterative and incremental (evolutionary) approach to software development
which is performed in a highly collaborative manner
by self-organizing teams within an effective governance framework
with "just enough" ceremony
that produces high quality software
in a cost effective and timely manner
which meets the changing needs of its stakeholders."

Typical development cycles of agile methods 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. A recent survey shows that agile methods have seen a far better success rate as compared to other methodologies [1].

Agile methods include many software development processes like Extreme Programming (XP), Scrum, Crystal methods, etc. The focus of this thesis is only on XP as it was used in the case study (Section 2.4) for more than two years. The interested readers can look at recent work of Dyba and Dingsoyr [37] who present a good review on the empirical studies of agile software development methods. The bibliography also presents good references to other agile methods.

1.1.1 Extreme Programming

One well known method from the group of agile software development methods is Extreme Programming. 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[1]. The result was the XP methodology. 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 [17][18]. In the second edition of his book, Beck added 'respect' as the fifth value [20].

- Improve communication: Improper communication in software teams is one of the root causes of failures within projects. Results are schedule slips, botched requirements, faulty development assumptions, and the like. XP addresses this problem by stressing good communication between all project stakeholders - customers, team members and project managers - on a consistent basis. A representative from the customer should be present on site at all times to answer questions and clarify project requirements; thus allowing a tight cooperation with the customer and immediate feedback. Programmers are expected to work simultaneously in pairs, with each programmer reviewing the other's work.

- Seek simplicity: XP tries to keep software development simple by allowing the programmer to concentrate only at the current task. This is expressed by the following slogans: "Do the Simplest Thing That Could Possibly Work" and "You Aren't Going to Need It" (known as YAGNI). The result is a more focused approach. The programmer can implement what he has to at that moment without thinking about future expansions. Even if it is clear now that some code has to be put in a few minutes later, it is not done yet if it is not needed for the current task. Please note that this is in total opposition of traditional approaches to software engineering!

- Get feedback on how well you are doing: A basic idea of XP is that there should always be some running system that delivers information about itself in a reliable manner. Feedback serves as a catalyst for change and an indicator of a project's progress. Code refactoring is derived from this value.

- Be able to proceed with courage: XP acknowledges that projects are ultimately people centric. It is the ingenuity of people, and not of any particular process, that causes projects to succeed. Courage also manifests

[1] http://www.computerworld.com/softwaretopics/software/appdev/story/0,10801,66192,00.html

itself in the features of feedback and refactoring, as described in the XP principles.

- Respect: Including all the stakeholders, the contributions of each person on the team need to be respected.

XP offers a number of practices, values and principles which are advised to be adopted in order to run a software development project [19]. XP is being experimented in different ways to make it fit to the specific needs of the projects as well as the development teams [135]. Being a lightweight agile method, XP has advantages of: on-time delivery, co-located team, relying on the team members' knowledge rather than documentation, optimized resource investments, short release cycles, working high quality software, tight customer integration, incremental design, constant communication and coordination, rapid feedback, continuous refactoring, pair programming, and test driven development [18], [20], and [7]. "XP is a collection of well-known software engineering practices. XP aims at enabling successful software development despite vague or constantly changing software requirements. The novelty of XP is based on the way the individual practices are collected and lined up to function with each other" [7]. It is also a people-oriented process with many social activities.

1.2 User-Centered Design

"Usability measures the quality of a user's experience when interacting with a product or system-whether a web site, a software application, mobile technology, or any user-operated device; User-centered design (UCD) is an approach for employing usability" [137]. Usability is defined by ISO 9241-11 [76] as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." Usability is a combination of many factors of a user interface that include ease of learning, efficiency of use, memorability, error frequency and severity, and subjective satisfaction [137]. UCD, also called human-centered design, is an approach to user interface design which is based on information about the people who will use the product. UCD process focuses on users through the planning, design, and development of a product [141]. ISO 13407 (Human Centred Design Process for Interactive Systems) [75] "is a description of best practice in user centred design. It provides guidance on design activities that take place throughout the life cycle of interactive systems. It describes an iterative development cycle where product requirements specifications correctly account

for user and organisational requirements as well as specifying the context in which the product is to be used. Design solutions are then produced which can be evaluated by representative users, against these requirements. The goal of the standard is to ensure that the development and use of interactive systems take account of the needs of the user as well as the needs of the developer and owner... to name but a few stakeholders [3]". ISO 13407 [75] defines four principles of Human-Centred Design [3]:

- active involvement of users
- appropriate allocation of function to system and to user
- iteration of design solutions
- multi-disciplinary design

It also defines four Human-Centred Design activities:

- understand and specify the context of use
- specify the user and organisational requirements
- produce design solutions
- evaluate designs against requirements

This standard was later improved in 2000 and was called ISO TR 18529.

Gulliksen et al. [47] propose a definition for user-centered system design (UCSD) with twelve principles: "UCSD is a process focusing on usability throughout the entire development process and further throughout the system life cycle". The twelve principles are: user focus, active user involvement, evolutionary systems development, simple design representations, prototyping, evaluate use in context, explicit and conscious design activities, a professional attitude, usability champion, holistic design, process customization, and a user-centred attitude [47].

Holzinger [57] emphasizes that every software practitioner should be aware of various usability engineering methods and apply them according to specific situations of a project. He provides details about two types of methods: inspection methods (heuristic evaluation, cognitive walkthrough, and action analysis) which do not require end-users for the evaluation but only UCD professional evaluates the system against standards, and test methods (thinking aloud, field observation, and questionnaires) which require end-users for the evaluation of system [57].

1.3 Motivation for Integrated Agile User-Centered Design Process

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 [68]. Another important factor for the success of a software application is user acceptance. "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" [94]. Although agile methods demand the role of the customer for continuous and close communication regarding project requirements, it is not necessary that this customer should be an end-user [91] [28]. Jokela and Abrahamsson [83] provide evidence that shows that the agile method "Extreme Programming" lacks usability, and the close coordination with the customer does not ensure the good usability of the product.

Some experts doubt that the XP process leads to true user-centered design [61]. The issues arising from this problem statement suggest that XP and UCD would not fit. But this perception is simplistic and misguided which has already been shown in practice where success is reported [93]. We can see succeeding practitioners combining usability/UCD and XP/agile methods by varying approaches [109], [28], [58], [93], [59], [133], [39], [40].

1.3.1 Why do some XP projects fail when including Human-Computer Interaction (HCI) work?

The following issues can prevent the integration of HCI instruments into XP processes [145]:

- Ad-hoc Input: Because of the short release cycles software engineers would need ad-hoc usability input during development. In practice, usability 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 psychological background, 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 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 [91] 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 humans they develop for.

- Awareness: In order to successfully include usability and user-centered design in an XP process the developers need to have a basic understanding of usability issues.

- Patchwork Experience: Our work-experience suggests that users often experience software as patchwork when developed from bottom up like done in iterative processes.

However, focus of both methodologies on users makes it possible to integrate them [45]. The integrated process allows to combine benefits of both methodologies and makes it possible to reduce the shortcomings of each because XP needs to know its true end-users and UCD benefits from a flexible and adaptive development methodology which runs throughout the project life-cycle [68].

1.3.2 Similarities between XP and UCD

The core values of XP and UCD [47] can be 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 interface. However, when comparing some of the core values it seems obvious that the two development processes can benefit from each other's practices.

End-user

One of the core practices of XP is to have a *Customer on Site* who is co-located with the programmers in order to answer domain-specific questions and give feedback on the system. This practice can be matched well with the testing of prototypes with actual users as proposed by UCD if the customer is also the real end-user or when developers are directly exposed to end-users.

Continuous 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*: in the first run, 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.

Iterative development

Both XP and UCD propagate an iterative procedure [47] [18] of design and development [141]. 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 cycles 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 refactoring is 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.

1.4 Problem Statement

As evidence showed that agile methods lacked usability awareness in their development lifecycle [83], and the integration of usability/user-centered design into agile methods was not adequately addressed. This reason motivated to the research work presented in this thesis that is to investigate the relationship between agile software development methods, particularly Extreme Programming and usability/user-centered design, how successfully these important methodologies are being integrated in industry, and interestingly how one can successfully integrate them. This leads to further interesting questions: How the role of a UCD professional[2] is carried out in organizations? What types of HCI techniques are used in teams/projects employing agile methods? To what degree are agile development methods and usability/UCD perceived by practitioners as being integrated and added value to their adopted processes and teams? The research methods that were used for this research work are: an online survey (Chapter 3), interviews from several agile UCD teams and analyzed through Grounded Theory (Chapter 4), and a case study of the m3 (mobile multimedia) project which was carried out for more than two years where several HCI techniques were introduced in the Extreme Programming process (Section 2.4 and Chapter 5).

1.5 Major Results of this Thesis

The successful integrated process of Extreme Programming and User-Centered Design methodologies

The integrated agile UCD process was successfully used for more than two years (Chapter 5) where a mobile multimedia application was developed. The HCI techniques that were integrated into the Extreme Programming agile method were: user studies, extreme personas, usability expert evaluations, usability tests, automated usability evaluations in the form of extended unit tests, and

[2]We use the term UCD professional equivalently to usability engineer, interaction designer, UI designer, user experience designer, etc.

both high and low-fidelity paper prototypes. User studies contained laddering interview techniques by means-end theory and field trial. The field trial included diary studies, contextual interviews, and focus group. The log file analysis of the field trial (Chapter 6) was also performed to know the viewing behavior of users for video content on mobile phones. All the problems that were discussed in Section 1.3.1 were addressed in the case study project. The tips and techniques for the practitioners, the solutions, and the lessons learned are presented in Chapter 5.

The mentioned agile UCD process can be successfully introduced in the projects having similar context. Particularly, the use of low-fidelity prototyping, usability expert evaluations, and rapid iterative testing easily fit within the fast moving iterations of agile methods. This is further supported by conducting a worldwide online survey (Chapter 3) and interviews from several agile UCD teams that were analyzed through Grounded Theory (Chapter 4). The results show that the majority of the respondents of the survey and the interviewed teams have dedicated UCD professionals mostly co-located. Where there is no dedicated UCD professional present in teams, this role is performed by developers having an interest in HCI. The top most used HCI techniques are low-fidelity prototyping, conceptual designs, observational studies of users, usability expert evaluations, field studies, personas, rapid iterative testing, and laboratory usability testing. The practitioners perceive that the integration of agile methods with usability/user-centered design has added value to their adopted processes and to their teams, has resulted in the improvement of usability and quality of the products developed, and has increased the satisfaction of end-users of the products developed.

The successful mobile multimedia streaming application

The mobile multimedia streaming application was successfully developed by employing the integrated agile UCD process. This is proved in the field trial and usability results of the application that show that the usability and the overall user experience of the developed application has significantly improved.

1.6 List of Publications

The research carried out for this work resulted into following publications:

1. (2007) User Interface Design for a Content-aware Mobile Multimedia Ap-

plication: An Iterative Approach [64]

2. (2008) User Interface Design for a Mobile Multimedia Application: An Iterative Approach [65]
3. (2008) Optimizing Extreme Programming [62]
4. (2008) Inside View of an Extreme Process [127]
5. (2008) Probing an Agile Usability Process [145]
6. (2008) Integrating Extreme Programming and User-Centered Design [67]
7. (2008) Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application [68]
8. (2009) Integration of Extreme Programming and User-Centered Design: Lessons Learned [69]
9. (2009) Concept and Design of a Contextual Mobile Multimedia Content Usability Study [66]
10. (2009) Current State of Agile User-Centered Design: A Survey [71]
11. (2009) Investigating Agile User-Centered Design in Practice: A Grounded Theory Perspective [72]
12. (2009) Practical usability in XP software development processes [63]
13. (2010) Analyzing real mobile web usage of a multimedia streaming application through log files [70]
14. (2010) Agile software development methods and usability/user-centered design: Perspectives from a survey [73]

1.7 Structure of this Thesis

The thesis is divided into chapters. Almost each chapter is a modified version of one or more publications. A short description of the chapters is presented here.

The present introductory chapter describes agile software development methods and user-centered design. It also presents an overview about the

problems of the integration of agile user-centered design, the problem statement and the motivation for carrying out the research for this thesis. This chapter contains excerpts from almost all the mentioned publications.

Chapter 2 presents the research methods used into the current work, namely the survey, interviews using Grounded Theory, and our case study project. It also presents the overview of the mobile multimedia application that has been developed within the context of this research. This chapter contains excerpts from the publications [71], [72], [73], and [70].

Chapter 3 describes in detail the survey carried out by the agile and UCD practitioners, worldwide. This chapter is a modified version of the publications [71] and [73].

Chapter 4 presents the interviews carried out from various agile UCD teams and the Grounded Theory used for the analysis. This chapter is a modified version of the publication [72].

Chapter 5 presents the case study where the work of the research carried out and agile UCD methodology was introduced. The approach integrates Extreme Programming process with HCI instruments: user studies, extreme personas, usability expert evaluations, usability tests, automated usability evaluations in the form of extended unit tests, and both high and low-fidelity paper prototypes. This chapter contains excerpts from the publications [64], [65], [145], [62], [67], [68], [69], and [63].

Chapter 6 presents the results of the log files analysis of the field trial carried out for the mobile multimedia application. This chapter is a modified version of the publication [70].

Finally, Chapter 7 concludes this thesis with future work directions.

Chapter 2

Method

This chapter describes the research methodologies that were followed for the scientific work of this thesis/research: survey, interviews using Grounded Theory, and a case study of our project.

2.1 Introduction

The research method for any scientific experiment is very important and should be clearly defined. The research methods that were used during this work are: survey, interviews using Grounded Theory, and a case study of our project called m3.

2.2 Survey

One of the research methodologies that were used for this work was a survey which is a method to collect information from a sample for learning something about a particular phenomenon or subject of interest. For this work an online web-based survey methodology was used while covering both quantitative as well as qualitative questions. The questionnaire was designed containing both close-ended multiple-choice questions and open-ended questions. A 5-point Likert scale was used for the close-ended multiple-choice questions; additionally don't know/no answer option was also provided. There were 28 questions ranging from demographic questions to agile methods and practices, as well as HCI techniques and the impact of the integration of agile methods and usability/UCD. For the validity, usefulness, and readability of the survey content, feedback was received from two of the pioneers and experts in the field of agile

usability/UCD, and then it was posted for data gathering. The details are given in Chapter 3. The survey questionnaire is presented in the Appendix.

2.3 Interviews using Grounded Theory

Besides technical focus, software development is mainly a human activity carried out by team members, so a qualitative approach is needed to study this human behavior. As with a qualitative approach, the researcher is forced to delve into the problem's complexity thus achieving richer and more informative results [125]. As the purpose was to study how the integration of agile methods and user-centered design is carried out in practice, which and how team members are integrating and using usability/UCD techniques into agile methods, the best fit was to use the grounded theory qualitative research method [44]. The grounded theory method is used to discover theory from data [8] - mostly from the 'voices' and 'experience' of the practitioners [30], i.e., interviews but also from other forms of data, such as observations and documents; and its application to human behavior is well-known [30]. Grounded theory is "useful for discovering behavioral patterns that shape social processes as people interact together in groups" [8]. It is used to generate a mid-level substantial theory that is directly derived from the data rather than verifying prior hypotheses [8]. Coleman and O'Connor [30] describe the analytical process of grounded theory, "The analytical process involves coding strategies: the process of breaking down interviews, observations, and other forms of appropriate data, into distinct units of meaning, which are labelled to generate concepts. These concepts are initially clustered into descriptive categories. They are then re-evaluated for their interrelationships and, through a series of analytical steps, are gradually subsumed into higher-order categories, or one underlying core category, which suggests an emergent theory." Basically for social sciences and specifically in nursing, grounded theory has also been applied in information systems, software engineering as well as in agile software development [50][30][144][39][40][27][41][53].

Grounded theory consists of a set of established procedures and guidelines for the systematic collection and analysis of qualitative data [144]; of which the constant comparison method is the heart of grounded theory [8], where data and concepts are constantly compared to each other during collecting and analyzing the data, to ensure that an integrative theory is developed which is grounded in the raw data [144]. In theoretical sampling, the sampling is

continuously selected based on emerging categories and concepts [44]. So the interview questions are being changed according to these emerging categories as well as "the researcher may decide to interview certain types of individual or seek out other sources of data" [30]. Further details are presented in Chapter 4.

2.4 Case Study

We have been working on a project called m3: mobile multimedia, in which we have been developing a multimedia streaming application for mobile phones. The project which started in summer 2007 is based in Austria. The project has several stakeholders and partners who come from various domains, including UI design, usability research, telecommunication and mobile network operators, content providers, and system integrators. This project's main scientific and academic goal is the analysis of agile software development methods and their integration with usability/user-centered design methodology. Another goal is the research in the field of mobile application usability. Additionally, the business partners are interested in the commercial aspects of the project. The team consists of six full-time regular members, five developers and a product manager who plays the role of the "on-site customer". One dedicated off-site usability engineer of a partner usability research center is also included in the team regarding usability guidance.

The famous agile method - Extreme Programming has been used as a process to carry out the application development. In XP, several HCI techniques have been integrated to develop a useful and usable product. The HCI techniques that were integrated into the Extreme Programming agile method are: user studies, extreme personas, usability expert evaluations, usability tests, automated usability evaluations in the form of extended unit tests, and both high and low-fidelity paper prototypes. Chapter 5 describes further details about the integrated agile UCD process and the lessons learned.

2.4.1 The Application

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 [36]. Still, there are not many full-featured applications on the market which utilize the available bandwidth and are accepted by the users.

The application that has been developed 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 contents available and unsatisfactory user-interfaces for accessing it. Usability is the key success factor for such applications [65][68].

The application consists of three main features: Home, Channel, and Category pages [70]. The Home page shows the most recent and top-rated clips for simple access, and provides basic search in the whole archive. It also provides links to Feedback and Help pages, where user can give feedback or get access to FAQ. Due to the native users, the application's interface is in German. Figure 2.1 shows the home page of the application.

Figure 2.1: Home page of the application.

The Channel page allows selecting a single channel and browsing its content by date and time, and to search for clips inside a channel. The Category page similarly allows to browse by categories and to search within a single category. When selecting a clip in the user interface, the clip's Detail page is displayed. It contains title and description of the clip, and provides access to clip comments

written by other users. Also, it provides means to rate the clip and write comments. Furthermore, a list with recommended clips is displayed.

The content is structured in terms of "channels". A channel can represent a real TV or radio broadcast channel, or it can be grouped thematically. Clips inside a channel are ordered by time, e.g., by the broadcast time for TV channels. This concept resembles the traditional TV schedule and is thus intuitively understandable by most users. However, in contrast to TV programs, also past programs can be browsed and watched. User has also an option to view content alphabetically. Additionally, all clips have a category. The category concept acts as a second dimension of structuring the content. Clips from different channels can have the same category (e.g., there are "News" clips in different TV channels). This allows a more fine-grained clustering of the content. Like channels, categories are a well-known concept for most users. Channels and categories help to coarsely browse through the clips, but for large archives users need more sophisticated tools to find their desired content. Therefore, the system also supports searching in the archive. Search can be restricted to a single channel or category, or be applied to the whole archive. The search functionality facilitates all available (mostly textual) metadata: the title of the clip, the description text (if available), and the creation/broadcast date. Additionally, a speech-to-text engine is used to generate transcriptions of spoken words, allowing the actual content of video and audio media to be searched.

The application is intelligent enough that it shows the context of spoken words in the clip so that user can immediately watch the particular part of the clip where these words were spoken or has the choice to watch the whole clip. With the search in the spoken text, it is possible to play-back the appropriate part of the audio and video clips directly from any of the places where these references occurred in the clip. Additionally, a timeline with graphical representations of the positions as well as a context is provided. Thereby in every entry a couple of the words surrounding the search term are shown. Figure 2.2 shows the search-result-details: one can see right away all the places where the search found results. The clip can be played-back by selecting the respective graphical marker and starts at the point of the corresponding position. Since these words are recognized automatically out of the spoken text, the feature is only meaningful with certain types of content such as newscasts, documentaries, etc. The speech-to-text feature is not 100% accurate. However, in a large number of cases the quality of the search results is sufficient. The searchable metadata is further enriched with user-generated information, facilitating the principles of the now omnipresent Web 2.0. Users are incorporated actively

Figure 2.2: Search results page showing the context of the spoken words found in the clip and showing their position with graphical markers on a timeline of the clip.

and passively in the process of gathering this information. On the one hand, users can rate clips they watched and write comments about these clips. This information helps to order clips by their importance and relevance for the user community. On the other hand, the system can recommend clips based on analyzing the viewing behavior of users: if a user watches a clip that has been watched by a significant number of other users, other clips watched by these users might be of interest as well.

The prototypical implementation of the application uses the program of two TV broadcast stations as archive material. The video content is captured by a 24/7 recording system and is added to the archive automatically.

The application also covers the area of interactivity: apart from usual current Web 2.0 features such as ratings, comments to entire clips and automatic recommendations, the application gives its user the possibility to put comments into every place of a video while watching it, which then can be seen by other users. Of course these comments can be turned on or off individually. New comments will replace older ones. An interesting possibility arises

through some newer mobile phone models such as the Nokia N85, the Samsung I7110 or the Sony Ericsson W980 that have a radio *transmitter* incorporated. In combination with the application, this enables the play-back of any archived radio transmission during, e.g., the trip to work in one's car on any standard car radio. Corresponding tests with the application were very successful.

The application is both substantially different compared to DVB-H based mobile phone TV and to UMTS/HSDPA based broadcast TV. It works with many current multimedia mobile phones, e.g., the Nokia N96 whereby in advance no additional software must be installed on the mobile phone. Modern mobile devices usually provide a high-speed UMTS Internet access and a standard-compliant HTML Web browser. Therefore, the user interface of the application is implemented in terms of standard Web technologies. The UI is optimized for use on mobile devices. The page layout is narrow enough to avoid horizontal scrolling on small displays, and the color scheme uses high contrasts to assure good visibility even in bad lighting conditions (e.g., direct sun light when using the application outdoor).

Chapter 3

Agile Software Development Methods and User-Centered Design: Perspectives from a Survey

This chapter investigates the relationship between agile development methods and usability/user-centered design. Agile methods are being adopted at an increasing rate in industry every year. However, there is evidence that shows that these methods lack usability awareness in their development lifecycle. So focus on the integration of these two important disciplines in software development - that is agile methods and usability/UCD - is increasing. To investigate how these important methodologies are being integrated we focused on three research questions; (1) How is the role of a UCD professional carried out in organizations? (2) What types of HCI techniques are used in teams/projects employing agile methods? (3) To what degree are agile development methods and usability/UCD perceived by practitioners as being integrated and added value to their adopted processes and teams? For this an online survey was conducted among UCD professionals and agile development team members which was responded by 92 participants throughout the world. In response to our first research question, our findings show that the majority of the respondents have dedicated UCD professionals mostly co-located in their teams. Above 10% responded that this role is performed by developers having interest in HCI/Usability, where there is no dedicated UCD professional present in teams. It is disappointing to know that 15.22% responded that there is no role of a UCD professional present in

the teams at all. Regarding our second research question the survey revealed that the top most used HCI techniques are low-fidelity prototyping, conceptual designs, observational studies of users, usability expert evaluations, field studies, personas, rapid iterative testing, and laboratory usability testing. For third research question the practitioners perceive that the integration of agile methods with usability/user-centered design has added value to their adopted processes and to their teams; has resulted in the improvement of usability and quality of the products developed; and has increased the satisfaction of end-users of the products developed. The findings give hope and new prospective for better usability and quality of the products developed through integrated agile usability/UCD methodology as well as increase the hope that both communities of UCD professionals and agile practitioners can work even closer to create useful and usable products.

3.1 Introduction

The focus of this chapter is to investigate the relationship between agile development methods and usability/user-centered design (UCD) - the two important disciplines in software development and how they are effectively used as an integrated software development process in industry.

Due to their popularity, agile software development methods are being adopted at an increasing rate in industry. Recently, Dyba and Dingsoyr [37] presented a good review on the empirical studies of agile software development. However, agile methods are still lacking usability awareness in their development lifecycle, and the integration of usability/user-centered design into agile methods is not adequately addressed. Although agile methods demand the role of the customer for continuous and close communication regarding project requirements, it is not necessary that this customer should be an end-user [91] [69]. Jokela and Abrahamsson [83] provide evidence that shows that the agile method "Extreme Programming" (XP) lacks usability and the close coordination with the customer does not enusre the good usability of the product. Holzinger [57] points out the need for the awareness of various usability methods by software practitioners and their application according to the context of a project. Memmel et al. comment that "When usability engineering becomes part of agile software engineering, this helps to reduce the risk of running into wrong design decisions by asking real end users about their needs and activities" [95].

The efforts of integrating usability/HCI into software engineering have already been carried out for many years, e.g., IFIP WG 2.7/13.4 working group has been formed [28]. A recent work is compiled by Seffah et al. [126] in the form of a book containing chapters about various aspects of the integration of usability into the development process. However, since agile methods are a recent and emerging idea, there has not been much work carried out regarding the integration of usability/UCD into agile methods. The research carried out and presented in this chapter aims at filling this gap and presents the results of a recently conducted online survey regarding the current state of the integration of agile methods and usability/UCD. This research is the continuation work of our previous research where we have been working on a case study of a mobile multimedia application employing an agile usability/UCD process [145], [68], [69]. Besides this we have also conducted interviews with several teams working in agile usability/UCD areas and analyzed data using grounded theory [72]. The data for the survey was collected from 92 practitioners throughout the world via several mailing lists of agile practitioners and UCD professionals.

The details are described in the sections below. The next section thoroughly describes related literature studies and present our research questions. Section 3.3 describes details about the research method. Section 3.4 describes the results with discussion. Section 3.5 concludes the chapter.

3.2 Related Literature Studies

This section presents related work in two sub-sections: Related studies on agile usability/UCD in general and studies regarding surveys on agile methods and usability/UCD.

3.2.1 Related Studies on Agile Usability/UCD

In 2002, Kent Beck and Alan Cooper discussed the integration of XP, one of the popular agile methods, with interaction design and concluded that both approaches have strengths that can be integrated [102]. The focus of both methodologies on delivering value and on customers/users, as well as their iterative nature and continuous testing, make it possible to integrate them and reduce the shortcomings of each methodology, as agile methods need to know their true end-users and UCD benefits from a flexible and adaptive development methodology which runs throughout the project life-cycle [68]. Several studies exist that examine various aspects of the integration of agile methods

and usability/UCD. Patton [109] gives details about the way of integrating interaction design into an agile process. Recently, Patton [110] describes the twelve best practices for adding user experience (UX) work to agile development. In their ethnographic field study, Chamberlain et al. [28] have described a framework for integrating UCD into agile methods. Armitage provides the guidelines for designers to work within agile methods [13]. Hodgetts [54] reports about his coaching experience for the integration of user experience design with agile methods. McInerney and Maurer [93] interviewed three specialists for integrating UCD within agile methods and were able to report the success of these methods. Miller [98] describes her experience of parallel tracks of interaction designers and developers that are highly connected and interleaved so that the interaction designers were always one iteration ahead. Blomkvist [22] describes the core principles of agile methods and UCD and outlines a model for bridging agile methods and UCD. Sy [133] describes her company's process of integrating agile methods with UCD in detail. Using Grounded Theory qualitative method, Ferreira et al. [39][40] have investigated several projects for the integration of UI design and agile methods. Fox et al. [41] also conducted a Grounded Theory qualitative study and describe the integration of agile methods and UCD in the industry.

Approaches to integrating agile methods and HCI practices vary. Constantine and Lockwood [32] focus on models in their agile usage-centered design. Kane [84] suggests integrating discount usability with agile methods. Beyer et al. [21] describe how Contextual Design, a UCD method, fits with agile methods. Holzinger et al. [58] [59] presented the idea of extreme usability by combining XP and usability engineering and embedded their ideas into software engineering education. Meszaros and Aston [97] report the introduction of usability testing based on paper prototypes into agile methods. Lee [88] describes combining scenario-based design, a usability engineering process, into agile methods. Obendorf and Finck [106] have described the integration of XP and scenario-based usability engineering. Brown et al. [24] report on using various artefacts, such as stories, sketches, and lists between interaction designers and agile developers. Ungar [136] describes the benefits of introducing Design Studio into the agile UCD process. Broschinsky and Baker [23] report on the successful use of personas in the XP process. Ambler [11] discusses strategies for tailoring user experience into agile methods using agile model-driven development. Wolkerstorfer et al. [145] and Hussain et al. [68], [69] have reported on the integration of various HCI techniques, e.g., field studies, personas, usability tests, paper prototypes, usability expert evaluations, etc.,

into their agile process. In the report of the special interest group regarding agile user experience, Miller and Sy [99] have referred to uncovering the best practices for agile UCD. Budwig et al. [25] report about the experience of UX teams working in Scrum, one of the popular agile methods, and describe the challenges, issues, and the solutions that they implemented to resolve those issues. Sy [134] describes "a framework for creating multi-sprint designs and getting them implemented without violating the Agile taboo against big design". Lee et al. [89] have developed and implemented an integrated approach of agile usability called extreme scenario based design. Most recently Barksdale and McCrickard [16] report on their concept mapping approach which they applied in an agile software project and conclude that this approach has value as a visual tool in agile usability. Many of the studies mentioned above provide only anecdotal views and there is a need for quantitative as well as qualitative research in this area.

3.2.2 Related Studies regarding Surveys on Agile Methods and Usability/UCD

There are few survey studies that exclusively regard agile methods and cover various aspects including their effectiveness and the potential problems [114][116][122][129][123]. Recently, two surveys were conducted among agile professionals for evaluating the success factors in agile software development projects and practices [29],[100]. None mentioned usability/UCD. Ambler runs surveys on various IT topics including agile development methods almost every year [1] but not specifically to agile usability/UCD.

There are various survey studies regarding usability, usability professionals, and user-centered design dating back to 1993-94 [115],[118]. In the survey of Gunther et al. [48], the highest rated HCI techniques were usability testing followed by prototyping and heuristic evaluations. The survey results of Vredenburg et al. [139] show that UCD methods are gaining extensive acceptance in industry. Gulliksen et al. [46] conducted a survey of the usability profession in Sweden, showing that usability and user involvement has low priority in commercial projects. The highest rated HCI techniques were thinking aloud, lo-fi prototyping, interviews, field studies, and scenarios. In their survey study, Jerome and Kazman [81] point out the lack of coordination between developers and HCI practitioners. Surprisingly, heuristic evaluation was the least used technique. Ji and Yun [82] also conducted a survey in Korea among devel-

[1] http://www.ambysoft.com/surveys/

opers and usability practitioners, which not only showed differences between the type of output and customer requirements but also that practitioners were aware that usability/UCD methods have improved the usability of the developed product. In Switzerland, Vukelja et al. [140] conducted a survey among developers regarding the focus on design and development of user interfaces. Their results show that without the involvement of HCI practitioners, developers frequently develop user interfaces, and usability tests are rarely conducted. Zhou et al. [146] conducted their UCD survey in China showing that UCD methods can improve users' satisfaction and the competitiveness of the products developed. Recently in Norway, Bygstad et al. [26] conducted their survey regarding the integration of software development methods and usability. Their results show that usability testing is perceived to be less important than usability requirements, and companies believe that both software development methods and usability are integrated. Most companies use their own software development methods, followed by RUP, and Microsoft solution framework, while XP/agile methods were the least used methods. This study does not specifically focus on the integration of agile methods and usability. In a recent study, Dayton and Barnum [34] conducted two surveys regarding the impact of UCD within one company before and after moving to agile methods. Focusing mainly on usability testing, the results show that after transitioning to an agile process, the company becomes aware that the use of informal usability tests fits better with the agile process and that these are as effective as the formal usability tests conducted in a laboratory. This study mainly presents the results from a technical communicator's point of view, focuses on usability tests, presents views from just one company, and does not address other usability/HCI techniques. Nielsen Norman group [105] has conducted a survey study regarding agile usability / user experience . They report that low-fi prototype are mostly used, usability professionals work in a parallel track, and faster usability methods work fine. This report is not publicly available though the summary is available at http://www.nngroup.com/reports/agile/.

3.2.3 Research Questions

To the best of our knowledge, no survey study except one by the Nielsen Norman group [105], has been conducted which specifically addresses the integration of agile methods and usability/UCD, focuses on both developers and usability professionals working in agile methods throughout the world, the role of the UCD professional carried out, the HCI techniques used in agile methods,

and their impact on the increased quality/usability of the products developed as well as added value to the adopted processes and teams. Our research aimed at filling out this gap by posing the following research questions and then conducting a survey.

RQ1: How is the role of a UCD professional carried out in organizations?

RQ2: What types of HCI techniques are used in teams/projects employing agile methods?

RQ3: To what degree are agile development methods and usability/UCD perceived by practitioners as being integrated and added value to their adopted processes and teams?

The next section outlines the method used for the research questions.

3.3 Method

This study used the online web-based survey methodology while covering both quantitative as well as qualitative research methods. A questionnaire was designed containing both close-ended multiple-choice questions and open-ended questions. A 5-point Likert scale was used for the close-ended multiple-choice questions; additionally a don't know/no answer option was also provided. In total, there were 28 questions ranging from demographic questions to agile methods and practices, as well as HCI techniques and the impact of the integration of agile methods and usability/UCD. For the validity, usefulness, and readability of the survey content, feedback was received from two of the pioneers and experts in the field of agile usability/UCD. It should be noted that not all questions were answered by all survey respondents.

The survey was targeted at practitioners (both UCD professionals and developers) working in agile methods that integrate some HCI techniques, or where the role of a usability professional is practiced by someone in their agile team, or who have some usability/UCD awareness in their processes. The survey was posted and distributed to the agile-usability and XP Yahoo groups, the CHI mailing list, the Austrian HCI-UE group, the British HCI group, Agile and Agile Alliance groups of LinkedIn as well as Agile Methods group of Xing professional networks. The survey was implemented by using the open source survey tool "LimeSurvey". The survey was started in the second week of June 2009 and was closed after five weeks with 92 responses. Table 3.1 shows the various job titles of the respondents. The job title 'Other' includes a product manager, an analyst, a technical writer/usability, a business analyst, an

academic researcher in HCI, and a researcher/programmer/student.

Job title	Frequency	Percent
Executive / Director	14	15.22%
Project / Program Manager	12	13.04%
Developer / Software Engineer / Programmer	16	17.39%
Usability Engineer / UI/UX/Interaction Designer	33	35.87%
Consultant	11	11.96%
Other	6	6.52%

Table 3.1: The various job titles of the 92 respondents

Table 3.2 shows the location of the respondents.

Location	Frequency	Percent
Europe	42	45.65%
North America	35	38.04%
Australia & New Zealand	6	6.52%
South & Central America	4	4.35%
Asia	3	3.26%
Africa	2	2.17%

Table 3.2: Locations of the 92 respondents

Table 3.3 shows the total experience of the respondents in software development.

Table 3.4 shows the experience of the respondents in agile methods.

Table 3.5 shows the industry sector of the respondents.

Table 3.6 shows the total number of employees working in the organizations of the respondents.

Table 3.7 shows the total number of members working in the teams of the respondents.

3.4 Results and Discussion

This section presents the results and discusses them from the point of view of several research questions. The first part presents general information of the

Experience	Frequency	Percent
1 Year	4	4.35%
2 - 5 Years	19	20.65%
6 - 10 Years	21	22.83%
11 - 20 Years	31	33.70%
more than 20 Years	13	14.13%
No answer	4	4.35%

Table 3.3: Total experience of the 92 respondents in software development

Experience	Frequency	Percent
1 Year	21	22.83%
2 - 5 Years	47	51.09%
6 - 10 Years	11	11.96%
11 - 20 Years	5	5.43%
No answer	8	8.70%

Table 3.4: Experience of the 92 respondents in agile methods

Industry Sector	Frequency	Percent
Financial	4	4.35%
Construction	1	1.09%
Education	9	9.78%
Government	3	3.26%
Manufacturing	1	1.09%
Health	4	4.35%
Software	27	29.35%
Telecommunications	6	6.52%
IT Services	19	20.65%
Transportation	3	3.26%
NGO	2	2.17%
Real Estate	2	2.17%
Retail	1	1.09%
Other	10	10.87%

Table 3.5: Industry sector of the respondents

Range of Employees	**Frequency**	**Percent**
1 – 50	31	33.70%
51 - 100	6	6.52%
101 - 500	13	14.13%
501 – 5000	16	17.39%
more than 5000	25	27.17%
No answer	1	1.09%

Table 3.6: Total number of employees working in the organizations of the respondents

Range of Employees	**Frequency**	**Percent**
1 – 10	55	59.78%
11 - 20	19	20.65%
21 - 100	11	11.96%
101 – 200	2	2.17%
201 – 500	1	1.09%
more than 500	2	2.17%
No answer	2	2.17%

Table 3.7: Total number of members working in the teams of the respondents

respondents regarding agile methods and practices as well as communication tools used, and when HCI / usability evaluations occur in their projects.

Table 3.8 shows some of the agile practices adopted by and some of the roles carried out in the teams of the respondents. The results are not so much comaparable with [114], [122], and [123] as focus of these surveys was on only one agile method: XP. But the results are comaparable with various survey results of Ambler [10], [12] and our results are slightly different than those results. For example the pair programming practice is rated as 36% most effective [12] whereas in our case it is rated as 41.30%.

Agile Practices & Roles	**Frequency**	**Percent**
High Level Release Planning	52	56.52%
Iteration / Sprint Planning	71	77.17%
Daily Stand-up / Scrum Meeting	61	66.30%
Sit Together	36	39.13%
Pair Programming	38	41.30%
Test Driven Development	50	54.35%
Refactoring	47	51.09%
Continuous Integration	43	46.74%
Automated Acceptance Testing	27	29.35 %
Collective Code Ownership	34	36.96%
Coding Standards	44	47.83%
Sustainable Pace	29	31.52%
Simple Design	48	52.17%
Stories	65	70.65%
Incremental Design	50	54.35%
Small Releases	42	45.65%
Retrospectives	46	50%
Status Reports	19	20.65%
On Site Customer Available	34	36.96%
Customer is an End-User	29	31.52%
Active Business Analyst Product Owner Customer Participant	45	48.91%
Agile Coach Present in Team	23	25%

Table 3.8: Some of the agile practices adopted by and some of the roles carried out in the teams of the respondents

Scrum is highly used among various agile software development methods followed by Extreme Programming (XP), proving the consistency of their growing adoption in industry. Table 3.9 shows the various agile methods used. Note that multiple answers were possible to select. The 'Other' option in agile methods contains TSP/Agile Fusion, agile UCD, and home grown methods within the company.

Method	**Frequency**	**Percent**
Scrum	62	67.39%
Extreme Programming (XP)	44	47.83%
Lean Development	17	18.48%
Agile Unified Process/ Open UP	9	9.78%
Pragmatic Programming	7	7.61%
Crystal Methods	5	5.43%
Adaptive Software Development	3	3.26%
Other	11	11.96%

Table 3.9: The various agile methods used (multiple answers possible)

We also asked in the survey which communication channels for conveying information with a UCD professional are better? The results reveal that the respondents prefered Face-2-Face communication followed by teleconference calls, online chat, video conferencing, and email. The results are almost consistent with Ambler[2] [10]. For example the report mentions that Face-to-Face communication with stakeholders is 60.5% "very effective" while our results show that it is 71.74% "very effective". Tables 3.10, 3.11, 3.12, 3.13, 3.14 show the ratings of the respondents.

An interesting question we also asked: when HCI / usability evaluations occur in the projects, over 90% responded that it occurs during all the phases of a project lifecycle. About 84% said that evaluations occur before a project begins as shown in Table 3.15. The results are consistent with [40] and [39].

In continuation, we also asked them: Do the requirement details emerge over time on your projects or are they fixed up-front? Over 92% responded that the requirement details emerge over time as shown in Table 3.16. The results are consistent with [40] and [10].

We also asked the respondents if the requirement details emerge over time then do they do some UI design up-front before any development begins in

[2]http://www.ambysoft.com/downloads/surveys/PracticesPrinciples2008.pdf

Face-2-Face communication	Frequency	Percent
Very Effective	66	71.74%
Effective	12	13.04%
Neutral	0	0%
Ineffective	0	0%
Very Ineffective	1	1.09%
Don't Know / No Answer	13	14.13%

Table 3.10: Ratings of the Face-2-Face communication channel for conveying information with a UCD professional

Teleconference Calls	Frequency	Percent
Very Effective	9	9.78%
Effective	31	33.70%
Neutral	21	22.83%
Ineffective	10	10.87%
Very Ineffective	4	4.35%
Don't Know / No Answer	17	18.47%

Table 3.11: Ratings of the Teleconference Calls communication channel for conveying information with a UCD professional

Online Chat	Frequency	Percent
Very Effective	4	4.35%
Effective	33	35.87%
Neutral	26	28.26%
Ineffective	6	6.52%
Very Ineffective	2	2.17%
Don't Know / No Answer	21	22.83%

Table 3.12: Ratings of the Online Chat communication channel for conveying information with a UCD professional

Video Conferencing	Frequency	Percent
Very Effective	10	10.87%
Effective	24	26.09%
Neutral	17	18.48%
Ineffective	5	5.43%
Very Ineffective	0	0%
Don't Know / No Answer	36	39.13%

Table 3.13: Ratings of the Video Conferencing communication channel for conveying information with a UCD professional

Email	Frequency	Percent
Very Effective	5	5.43%
Effective	26	28.26%
Neutral	24	26.09%
Ineffective	17	18.48%
Very Ineffective	6	6.52%
Don't Know / No Answer	14	15.22%

Table 3.14: Ratings of the Email communication channel for conveying information with a UCD professional

Answer	Percent
Before the project begins	84.44%
During the project (all phases)	90.78%
During the project (acceptance test phase)	82.05%
Missing	7.61%

Table 3.15: when HCI / usability evaluations occur in projects (multiple answers possible)

Answer	Percent
Emerge over time	92.85%
Fixed up-front	7.15%

Table 3.16: Comparison between the requirement details whether they emerge over time on projects or whether they are fixed up-front

projects? As can be seen from Table 3.17, more than 88% responded in affirmation. The results are comaprable with [40] and [10] and are consistent.

Answer	Percent
Yes	88.88%
No	11.12%

Table 3.17: If the requirement details emerge over time then whether some UI design up-front is done before any development begins in projects

3.4.1 RQ1: Role of UCD Professional

Table 3.18 shows how the role of a UCD professional is carried out in the organizations of the respondents. The majority of the respondents have dedicated UCD professionals - mostly co-located in their teams. Above 10% responded that this role is performed by developers having interest in HCI/Usability, where there is no dedicated UCD professional present in teams. These results are consistent with [39], [40], [41]. It is disappointing to know that 15.22% responded that there is no role of a UCD professional present in the team at all. The Other option describes various situations of the respondents as described by them: *"usability engineer - location varies", "sometimes co-located, central at others", "Interaction designer co-located, usability engineer not co-located", "BA with interest does this", "special UI team at one site (we have several sites)", etc.*

Role of UCD professional carried out	Frequency	Percent
UCD professional co-located with team	42	45.65%
UCD professional not co-located with team	16	17.39%
Developer having interest in HCI/Usability performs it	10	10.87%
No role of a UCD professional present in the team at all	14	15.22%
Other	10	10.87%

Table 3.18: The role of a UCD professional carried out in the organizations of the respondents

In continuation of the previous question, we also asked them If a UCD professional is present in your team, in your opinion which approach is better:

co-located vs. remotely located UCD professional? As can be seen from Table 3.19, the majority prefered the the role of co-located UCD professional.

Which approach is better	**Percent**
Approach of co-located UCD professional is better	95.58%
Approach of remotely located UCD professional is better	4.42%

Table 3.19: Comparison of the approach of co-located versus remotely located UCD professional

We also asked them whether dedicated UCD professional present in a team is a better approach or this role performed by a developer having interest in HCI/usability. As can be seen from Table 3.20, the majority prefered the approach of having a dedicated UCD professional present in the team. However, many respondents commented that if there is no dedicated UCD professional present in the team then a developer having interest in HCI/usability is also a good substitute. Consequently, we also asked whether the respondents think that it is possible that a developer can pick-up HCI skills. As can be seen from Table 3.21, the response was positive in the sense that a developer can pick-up HCI skills where almost 50% said definitely/nearly, while 40% said it is possible sometimes. In the next consecutive question we asked how a developer can learn HCI skills. The response is shown in Table 3.22. The Other textfield provided many interesting suggestions from the respondents: *"Study and learning by doing", "usability testing designs with users as wells as acceptance testing", "Research, practice, intuition", "Self learning", "must have personal interest & aptitude".*

Which approach is better	**Percent**
Having dedicated UCD professional	92.85%
This role performed by a developer having interest in HCI/usability	7.15%

Table 3.20: Comparison of the approach of having dedicated UCD professional versus this role performed by a developer having interest in HCI/usability

3.4.2 RQ2: HCI Techniques Used:

As can be seen from Table 3.23, the top most used HCI techniques are low-fidelity prototyping, followed by conceptual designs, observational studies of

Answer	Percent
Definitely	39.53%
Nearly	9.30%
Sometimes	40.69%
Rarely	9.30%
Never	1.16%

Table 3.21: Whether it is possible that a developer can pick-up HCI skills

Suggestions	Frequency	Percent
By Pairing with a UCD Professional	75	81.52%
By Training	66	71.74%
Other	15	16.30%

Table 3.22: How can a developer learn HCI skills? (multiple answers possible)

users, usability expert evaluations, field studies, personas, rapid iterative testing, and laboratory usability testing, respectively. Note that multiple answers were possible. The 'Other' option in HCI techniques used contains contextual inquiry, non-formal usability tests (in person), participatory design, thorough UI specifications, high-fidelity prototyping, and model-driven inquiry. The use of low-fidelity prototyping, usability expert evaluations, and rapid iterative testing easily fit within the fast moving iterations of agile methods. The results are slightly different from those of [118][48][139][46][82][146]. The use of low-fidelity prototyping, usability expert evaluations, and rapid iterative testing easily fit into the fast pace of agile methods. These results are mostly consistent with Nielsen Norman report [105].

3.4.3 RQ3: The Impact of the Integration of Agile Methods and Usability/UCD:

The majority of the respondents consider that the integration of agile methods with usability/user-centered design has added value to their adopted process and to their teams, as most have selected 'Strongly Agree' or 'Agree' options. Table 3.24 shows the answers in frequency and percent. Only 4 respondents have selected 'Disagree' or 'Strongly Disagree'.

We investigated this question further to find the difference of opinion be-

HCI Techniques	Frequency	Percent
Low-Fidelity Prototyping	63	68.48%
Conceptual Designs	55	59.78%
Observational Studies of Users	52	56.52%
Usability Expert Evaluations	47	51.09%
Field Studies	43	46.74%
Personas	41	44.57%
Rapid Iterative Testing	37	40.22%
Laboratory Usability Testing	36	39.13%
Needs Analysis	33	35.87%
Goal-Directed Design	27	29.35%
Remote Usability Testing	24	26.09%
Conceptual Inquiry	24	26.09%
Ethnographic Research	21	22.83%
Automated Usability Evaluations	7	7.61%
Other	11	11.96%

Table 3.23: The various HCI Techniques used (multiple answers possible)

Answer	Frequency	Percent
Strongly Agree	27	29.35%
Agree	40	43.48%
Neutral	7	7.61%
Disagree	2	2.17%
Strongly Disagree	2	2.17%
Don't know / no answer	14	15.22%

Table 3.24: The integration of agile methods with usability/UCD has added value to the adopted process and to the teams

tween UCD professionals and developers. A chi square value ($\chi^2 = 0.0692$; p=.05) indicates that there is no statistically significant difference between the opinions of UCD professionals and developers.

In Table 3.25, it can be seen that most respondents perceive that the adoption of the agile user-centered design process by their teams has resulted in the improvement of usability and quality of the product developed.

Answer	**Frequency**	**Percent**
Strongly Agree	19	20.65%
Agree	41	44.57%
Neutral	10	10.47%
Disagree	6	6.52%
Strongly Disagree	2	2.17%
Don't know / no answer	14	15.22%

Table 3.25: The adoption of an agile UCD process has resulted in the improvement of usability and quality of the product developed

In connection with the usability of the products, the majority of the respondents are also of the opinion that, due to the agile user-centered design process adopted by their teams, the resulting product has increased the satisfaction of its end-users (see Table 3.26). The results are consistent with [82] [146].

Answer	**Frequency**	**Percent**
Strongly Agree	20	21.74%
Agree	38	41.30%
Neutral	5	5.43%
Disagree	5	5.43%
Strongly Disagree	2	2.17%
Don't know / no answer	22	23.91%

Table 3.26: The resulting product has increased the satisfaction of its end-users due to the adoption of an agile UCD process

3.4.4 Limitations

There are few limitations to this research. Although the survey was responded by 92 practitioners which is a good number of response, the results cannot be

generalized to the entire world as every project has its own context. Despite this we believe that the results presents a good picture of the field as the survey was responded by real practitioners throughout the continents.

We also acknowledge that the survey questionnaire could have been improved: we should have asked about the attitudes and behaviors of the UCD professionals and the developers as they both are from different backgrounds so it would have been of interest how they work together in agile teams. But we thought that it would increase the considerable size of the questionnaire so we have planned for a separate questionnaire which specifically focuses on the behavioral issues.

We have not much focused on the statistical significance analysis due to the simplicity of our data so we presented the results mostly in the form of summaries.

3.5 Conclusion

This chapter investigated the relationship between agile development methods and usability/user-centered design through an online survey. Agile software development methods are flexible, iterative, and lightweight, making it easy to integrate usability/HCI techniques into them, while the focus of both methodologies on delivering value and on customers/users, as well as their iterative nature and continuous testing, further facilitate and enable this integration [68]. The survey results support this as the majority of the respondents perceive that the integration of agile methods with usability/user-centered design has added value to their adopted process and to their teams. They also perceive that the adoption of an agile user-centered design process by their teams has resulted in the improvement of usability and quality of the product developed and has also increased the satisfaction of its end-users.

The majority of the respondents have dedicated UCD professionals mostly co-located in their teams. Few responded that this role is performed by developers having interest in HCI/Usability, where there is no dedicated UCD professional present in team.

The top most HCI techniques used are low-fidelity prototyping, followed by conceptual designs, observational studies of users, usability expert evaluations, field studies, personas, rapid iterative testing, and laboratory usability testing, respectively. The use of low-fidelity prototyping, usability expert evaluations, and rapid iterative testing easily fit into the fast pace of agile methods. Other

techniques can be adapted using two parallel tracks of interaction designers and developers [133].

The results are promising and increase the hope that both communities of usability professionals and agile practitioners can work even closer to create useful and successful products so that the use of those products can be brought to their full potential for the betterment of society.

Chapter 4

Investigating Agile User-Centered Design in Practice: A Grounded Theory Perspective

This chapter investigates how the integration of agile methods and User-Centered Design is carried out in practice. For this study, we have applied grounded theory as a suitable qualitative approach to determine what is happening in actual practice. The data was collected by semi-structured interviews with professionals who have already worked with an integrated agile UCD methodology. Further data was collected by observing these professionals in their working context, and by studying their documents, where possible. The emerging themes that the study found show that there is an increasing realization of the importance of usability in software development among agile team members. The requirements are emerging; and both low and high fidelity prototypes based usability tests are highly used in agile teams. There is an appreciation of each other's work from both UCD professionals and developers and both sides can learn from each other.

4.1 Introduction

The adoption of agile software development methods is growing in the industry. However, in their development lifecycles, these methods still lack the realization of the importance of usability and usable user interfaces. Holzinger [57]

draws attention to the need for an awareness of various usability techniques by software practitioners that should be applied according to the nature of a project. Both the agile methods and user-centered design methodologies have many similarities: both methodologies focus on delivering value, both focus on customers/users, and their iterative nature and continuous testing are the key similarities for integrating them easily [68]. However, there has not been much investigation regarding how these two methodologies are actually practiced in industry; and how to successfully integrate HCI/usability techniques into agile methods is an area worthy of exploration.

In this chapter, we report our findings, the emerging themes of our qualitative research conducted using grounded theory [44]. The data was collected by semi-structured interviews with professionals who have already worked with an integrated agile UCD methodology. Further data was collected by observing these professionals in their working context, and studying their documents-the various artefacts produced during their work, where possible. All the interviews were conducted by me. We also confirm some of the previously identified themes by Ferreira et al. and Fox et al. [39][40][41].

The next section describes related literature studies. Section 4.3 provides details about the research method and participants' profiles. Section 4.4 describes the results – the emerging themes along with quotes from the participants. Section 4.5 concludes the chapter with future work.

4.2 Related Literature Studies

There have been studies present in the literature that report about the various aspects and efforts for the integration of agile methods and usability/user-centered design. Patton [109] reports about how to combine interaction design into his agile process and emphasized the participation of all stakeholders into the design process. His team was responsible for the UCD practices as there was no dedicated UCD professional present in the team. In his recent article, Patton [110] describes the twelve best practices for adding user experience (UX) work to agile development. Chamberlain et al. [28] propose a framework for integrating UCD into agile methods by presenting similarities between two methodologies. McInerney and Maurer [93] interviewed three UCD professionals for integrating UCD within agile methods and their report was positive. Holzinger et al. [58] presented the idea of extreme usability by integrating XP and usability engineering which then, they implemented into software en-

gineering education. At her company, Miller [98] reports her experience where both interaction designers and developers were working in parallel tracks coordinating their work smoothly. Meszaros and Aston [97] describe how they introduced paper prototype based usability testing into agile process. Memmel et al. describe the benefits of including usability engineering into agile development, "When usability engineering becomes part of agile software engineering, this helps to reduce the risk of running into wrong design decisions by asking real end users about their needs and activities" [95]. Sy [133] reports the parallel tracks used at her company by both UCD professionals and developers to integrate UCD with agile methods. At every iteration cycle, the coordination between UCD professionals and developers was smoothly done so that UCD professionals were always one cycle ahead, in order to be able to gather requirements and design for the next cycles while testing the previous cycle's work. Ferreira et al. [39][40] have investigated four projects in four different countries for the integration of UI design and agile methods. They have used the grounded theory qualitative method for their study where they have conducted semi-structured interviews from the two members of each project, one who concentrated on UCD and one who concentrated on programming. Some themes that emerged from their data are: there is an advantage in doing up-front interaction design; do most of up-front design; much of interaction design consists of studying clients and users; interaction design learns from implementation by developers; cost and time are constraints; both usability testing and development affect each other; and agile leads to changing the relationship of UCD professionals and developers. Using the same grounded theory qualitative approach, Fox et al. [41] also investigated the integration of UCD with agile methods. They conducted semi-structured interviews from ten participants, one member from each team, in North America and Europe where the majority of the participants were UCD professionals and only few were developers. They describe three approaches taken by the participants to achieve the integration and term these approaches as generalist, specialist, and hybrid approach. In the generalist approach, there is no UCD professional present in a team but this role is performed by a developer interested in usability/UCD. In the specialist approach, one UCD professional is present in the team; whereby in the hybrid approach, a team member has both formal UCD training and also has experience in software development. Brown et al. [24] report that they use various artefacts, such as stories, sketches, and lists between UCD professionals and developer in their agile process. Ungar [136] reports the advantages of introducing a Design Studio, a collaborative work-

shop, into the agile UCD process. "The design studio is a rapid process that allows designers, developers and stakeholders to collaborate and explore design alternatives. Participants grow their skills by exchanging viewpoints with their peers and openly discussing the strengths and weaknesses of their work. The design is enriched and strengthened from the feedback" [136]. In their work, Wolkerstorfer et al. [145] and Hussain et al. [68], [69] report on the integration of various HCI techniques, e.g., field studies, usability tests, paper prototypes, usability expert evaluations, etc., into their agile process. In their case study Federoff et al. describe the struggle of UX teams when transitioning to agile development [38]. Recently, Miller and Sy [99] discussed issues regarding agile user experience.

With a few exceptions, many studies are anecdotal, albeit providing important information how the UCD is integrated with agile methods at various levels and at various levels of efforts. Studying the various aspects of the integration of agile methods and usability/UCD is an area worth to explore for developing usable and quality software products. Our research is aimed at enhancing this knowledge base to identify new themes, and supporting and confirming the existing empirical evidence by using qualitative approach of grounded theory.

4.3 Method

Besides technical focus, software development is mainly a human activity carried out by team members, so a qualitative approach is needed to study this human behavior, as with qualitative approach, researcher is forced to delve into the problem's complexity thus achieving richer and more informative results [125]. As we wanted to study how the integration of agile methods and user-centered design is carried out in practice, which and how team members are integrating and using usability/UCD techniques into agile methods, we chose to use grounded theory qualitative research method [44]. The grounded theory method is used to discover theory from data [8] - mostly from the 'voices' and 'experience' of the practitioners [30], i.e., interviews but also from other forms of data, such as observations and documents; and its application to human behavior is well-known [30]. Grounded theory is "useful for discovering behavioral patterns that shape social processes as people interact together in groups" [8]. It is used to generate a mid-level substantial theory that is directly derived from the data rather than verifying prior hypotheses [8]. Coleman and O'Connor [30] describe the analytical process of grounded theory, "The analytical process

involves coding strategies: the process of breaking down interviews, observations, and other forms of appropriate data, into distinct units of meaning, which are labelled to generate concepts. These concepts are initially clustered into descriptive categories. They are then re-evaluated for their interrelationships and, through a series of analytical steps, are gradually subsumed into higher-order categories, or one underlying core category, which suggests an emergent theory." Basically for social sciences and specifically in nursing, grounded theory has also been applied in information systems, software engineering as well as in agile software development [50][30][144][39][40][27][41][53].

Since the initial discovery of grounded theory by Glaser and Strauss [44], there are now at least three versions of grounded theory [8] but we employed the Strauss and Corbin approach [130] and [131], because they argue that "the researcher's personal or professional experience is supportive of theory building and contributes to theoretical sensitivity, the ability to understand the data's important elements and how they contribute to theory" [30]. We have the experience of both as professionals as well as researchers in the study area, i.e., agile usability/UCD; and are already familiar with the literature, thus this knowledge supports theoretical sensitivity. Another reason for selecting the version of Strauss and Corbin is that "they favour setting the research question in advance of commencing a grounded theory study, rather than it being allowed to emerge at the coding phase as advocated by Glaser" [30]. We also set a few questions in advance: How the integration of agile methods and user-centered design is carried out in practice? Which HCI techniques are being used in agile methods? How the role of UCD professional is carried out in agile teams? Besides finding new themes, we also wanted to confirm the studies of Ferreira et al. [39][40] and Fox et al. [41].

Grounded theory consists of a set of established procedures and guidelines for the systematic collection and analysis of qualitative data [144]; of which the constant comparison method is the heart of grounded theory [8], where data and concepts are constantly compared to each other during collecting and analyzing the data, to ensure that an integrative theory is developed, which is grounded in the raw data [144]. In theoretical sampling, the sampling is continuously selected based on emerging categories and concepts [44]. So the interview questions are being changed according to these emerging categories as well as "the researcher may decide to interview certain types of individual or seek out other sources of data" [30]. We developed an initial interview protocol with the help from literature and from our own experience. Initially, we conducted a pilot interview with a project manager who is also a certified scrum master and

works in a software developing company on a project about social networking and is employing scrum agile method. We then adjusted our interview protocol by getting feedback from his interview and it was slightly but continuously modified as the interviews were conducted and data was gathered. For the analysis, we used Atlas TI software tool, which is suitable for grounded theory analysis. In grounded theory analysis, the first step is open coding, where codes or labels are allocated to data obtained from interview transcripts and from other sources, if any [30]. The purpose is to identify important concepts in the data and then categorize it [41]. During our open coding analysis, codes were allotted to data, which were then constantly compared to subsequent data. At the end of open coding, initial concepts and categories are identified, which are then used in axial coding to make relationships among categories and their sub-categories, and identifying core category/categories. We compared initial categories with subsequent categories and made relationships among them in the form of categories and subcategories. Finally, in selective coding, core categories are used to find themes or high-level concepts that emerge from the data [41]; and that explain how the participants are carrying out their work and solving their problems. Another ongoing technique in grounded theory is memoing which is "the ongoing process of making notes and ideas and questions that occur to the analyst during the process of data collection and analysis" [124]. These memos play an important role in identifying ideas or hypotheses and generating the themes or high-level concepts [30]. We continuously wrote memos during the interviews and during the analysis of the data, which helped us in generating the themes during all phases of the analysis and those emerging themes are described in the results section.

4.3.1 Participants

We denote the participants as P1, P2, P3, P4, and P5. Based in Austria, P1 is a usability expert who provides consultancy to both agile and non agile projects ranging from banking sector to transport to typical web based applications whose durations vary from 6 months to 3 years. He also delivers lecture at a local university. He works with the companies whom either do not have expert UCD professionals or have certain knowledge of usability but they do not have resources to manage it, so they have to buy resources. Depending upon the projects, sometimes he is the only UCD professional and sometimes his two other colleagues also work as UCD professionals in the projects. In one of the companies that is employing the XP process, he works as a UCD professional,

so according to Fox et al. he is performing the role of specialist [41].

Based in Austria, P2 is a developer who also has an interest in usability so he is performing the role of generalist [41]. He works in a team with 3 team members: one project manager and two developers. They use a combination of XP and Scrum and modified their practices according to their context. The web based product is for sales people to optimize their sale process.

P3 is based in Finland, and works in a large mobile phone company. He is a program manager and works on multiple projects using scrum. They develop mobile phones and telecom applications and also configuration tools for internal use. After failure on a project using the traditional waterfall process, he switched to agile methods and now he is continuously working on different projects by employing scrum. In one of the projects, the team size is about 6 people, one person as tester, 3 to 4 developers, and a UCD professional; so the role of specialist is being carried out in this team [41].

P4 is based in Germany and is an agile coach, a consultant and a certified scrum master. Mostly, he has been working as a project manager consultant with telecom sector and has been developing and leading teams for mobile phone applications. In one of the agile projects where he is working as a coach and project manager using scrum, there are 8 members who all are developers as well as having good background in usability as according to [P4], *"In that team, most of the people are actually coming from user interface design for mobile phones, so they almost all have a feel of what is a good UI because they saw many of those... They already have quite a lot experience of developing UI and some of them are developing these types of things something like 14 years but, one even 15 years, even before working in agile process. Usability was a part of our job description since quite long time... The general concepts about usability are discussed together by the whole team including management; and the low level usability, so what happens when the user presses a button, how for example scrolling goes, those are mostly decided by the engineers themselves."* The role of UCD professional in this case is carried out differently as all are both developers as well as having good experience in UI design, nevertheless, we would call this type of role a hybrid approach, as described by Fox et al. [41].

Based in Austria, P5 is a developer and project manager who has also experience in usability. He works on products which are used in hospitals for the medical staff and he also develops software for clinical research. They are using an agile process, which has been adapted to their context where many agile practices have been taken from pragmatic programming, scrum, and XP.

The team consists of 3 to 4 persons, among them one is a usability engineer who is not present in the team all the time but is accessible on demand. He is contacted for the formal usability tests, otherwise mostly the usability tests, i.e., low and high fidelity tests and thinking aloud tests are conducted by P5. Here again, the role of UCD professional is carried out differently as both the role of specialist as well as generalist is present in this project. So again, this is a new situation for the role of UCD professional compared to that described by Fox et al. [41].

4.4 Results

This section describes some of the main themes and concepts that emerged after the analysis of the data. The relevant passages from the interview participants are also presented.

Among agile team members, there is an increasing realization of the importance of usability in software development, whereas agile methods also provide advantages to usability/UCD for its integration.

[P1] *"This is also an indicator for me that it is good to use agile developments because then you have shorter cycles and then you have the possibility to integrate in the next cycle. If you use conventional project processes, you don't have the possibility, for example for half a year, to integrate. So agile development is good in that the usability results can be easily integrated in its short cycles and end users become more satisfied."*

[P2] *" We have no usability engineers, but I have interest in usability... Usability fixes can be done because of agile development we always have a working software so it is easy to find usability changes which then can be fixed according to the priority."*

[P3] *"The usability guy has value for the project... He thinks how the process and one sitting proceeds; so which functions are more used, which should be more easily available, which is used, which is the harder way to access, how to jump in from one component to other, and that kinds of issues. He has better vision and intuition what is better and what is not."*

[P4] *"Usability testing and taking care of UI into agile process, it is beneficial...The usability tests and evaluating UI with users bring lot of feedback to be fixed, so it was easy for us to fix those usability results into small iterations of agile process."*

[P5] *"Being both a developer and having interest, experience in usability is*

a plus point. If your product is usable, the user acceptance is far better than if it is only functional but unusable... Usability is always a main feature of the software, so the good usability is really important."

Almost all the participants mentioned that integrating usability/UCD into agile process has enhanced the quality and the usability of the product and increased the satisfaction of the users. Some also pointed out that this integration has added value to their team and the process.

[P1] *"We also have some numbers showing about the company that they compared page usage with user sessions and it doubled their usage and this was because of our usability work."*

[P2] *"Yes, we have good experience with that and customers, who are both customers and end users, are happy about that; they give us good feedback; they get what they want; and so we don't have bad experience about that."*

[P3] *"Yea, well, on this case I say, it's quite good proof because we tried on traditional way and it crashed and burned and nothing got. It was totally unusable, and this [using usability into agile process] we got... also the short iterations are good or the customer doesn't forget what he has asked for. Because in first case, they had changed their minds over the half year period, so that's why it wasn't any more what they expected and wanted. So now in agile, we can show the customer what he asks for immediately... The integration of usability techniques into agile method has added value to our team and the process. I definitely would like to use more."*

[P4] *"As the product is new and not yet public so we don't have real external users but it has increased the satisfaction of the internal users... Usability testing and taking care of UI into agile process, it is beneficial."*

[P5] *"Of course, yes."*

Almost all the participants mentioned that they do some up-front design to understand users, their goals, and the project vision; and the requirements are emerging. This result is consistent with other studies [39][40][41].

[P2] *"For the initial vision of the project we do paper prototyping with the customers [who are also end users] to understand their goals; and then we are getting continuously requirements from the customers during every cycle."*

[P3] *'Initially, we [as a team] did prototyping with end users, we discussed it with them and said this is what you are and on which you are comfortable and then on demo when we went through and we sat after wards and asked does it work for you? How are your feelings? ....The requirements are gathered for about one sprint. The usability guy takes feedback from the team members, from customers and user representatives, and shows them and asks them how*

they feel it; and of course there is feedback that this design works very well and this does not work and then it as taken as a requirement for the next sprint and so then it is fixed.‘”

[P4] *“The requirements are emerging and they are not fixed up front. As the product is completely new, so the customer had no clear requirements, he just provided few requirements and the majority of the requirements are generated internally by us [the team and the internal users at the company]; and there were few requirements also generated by technology. So the customer just had the vision of the product, a generic vision and we are putting meat into it [the features]. There was informal feedback got from outside but no formal feedback because they wanted to keep their product secret...Initially, paper prototype was something used in brainstorming and even in product manifest period and were shown internally. So in visionary meetings we used drawing boards, and also simulated them using NetBeans with GUI designer with the screen shots and how you move.”*

[P5] *“ The requirements are always emerging. The customer has only the vision. So only 20% to 30% requirements are already fixed and the rest are emerging. So the html mockup is the first version we implement and evaluate them to the customer to get more requirements.”*

There is an appreciation of each other's work from both UCD professionals and developers and both sides learn from each other. This result is also consistent with [39][40][145][69].

[P1] *“Because of different understanding, from developer perspective from usability perspectives, we are presenting the results in the form of workshop; so we try to invite developers and the other team members to discuss and to ask questions, because this is more efficient. During the discussion a developer pointed out technical, legal and security issues. And this was also an information for us; so we said ok we have to consider it in future.”*

[P3] *“Of course we have to consider how it is difficult to implement. So we have the conversation that what is possible at technical level and in this way the usability guy also got feedback from the developers. For example, the usability guy is presenting something and saying this is good and then when it is tested for example, in a testing environment and then it came out that it's response time is too long and on paper it looks good but it is not usable.”*

Both low and high fidelity prototype based usability tests are frequently conducted in the agile teams of these participants. The forms of prototypes vary and include paper prototypes, screenshots, powerpoint presentations, html mockups, etc. This result is also consistent with [97] and [69]. Holzinger [56]

describes the advantages and disadvantages of both low and high fidelity prototypes during their application in the development of a virtual medical campus interface. Usability tests on working software were also conducted by some of the participants. Other HCI techniques that were mentioned by the participants and sometimes also conducted are heuristic evaluations and thinking aloud. Fixing the usability feedback results by developers in an iteration cycle depends upon how big the usability change is. If it is big then it is implemented in the next iteration cycle, otherwise it is fixed in the same iteration cycle.

4.5 Conclusion

This chapter investigated how the integration of agile methods and user-centered design is carried out in practice using grounded theory. The emerging themes that the study found show that, among agile team members, there is an increasing realization of the importance of usability in software development; whereby agile methods also provide advantages to usability/UCD for its integration. The participants also mentioned that integrating usability/UCD into agile process has enhanced the quality and the usability of the product and increased the satisfaction of the users. Some also pointed out that this integration has added value to their team and the process.

Some up-front design is carried out to understand users, their goals, and the project vision. The requirements are emerging. The usability changes are fixed in the same iteration cycle or in the next iteration cycle depending upon how big they are. Both low and high fidelity prototypes based usability tests are highly used in agile teams. There is an appreciation of each other's work from both UCD professionals and developers and both sides can learn from each other. Almost all the results are consistent with the existing studies present in the literature. We also confirmed some of the previously identified themes by Ferreira et al. and Fox et al. [39][40][41].

It should be noted that the sample data set of five participants for grounded theory research is small which may be the limitation of the study but we plan to conduct more interviews from various participants working on different projects to verify and extend the results found during the current study. The current qualifying core category is "the realization of the importance of usability in software development", which will be verified by conducting further interviews and collecting various forms of data.

Chapter 5

Integration of Extreme Programming and Usability / User-Centered Design: A Case Study

This chapter describes the experiences made and lessons learned in our project. We investigate the potential of XP to produce usability-optimized products by incorporating HCI experts. We relate the software development method to user-centered design instruments and propose solutions to different usability 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, usability expert evaluations, and extended unit tests. We also provide short description of XP practices that we used in our project. The conclusion provides tips and tricks for practitioners.

5.1 Case Study

As mentioned in Section 2.4, we have been working on a mobile multimedia project where several HCI techniques have been integrated into the adopted Extreme Programming process. The details are described in the following sections.

5.1.1 Approach

The novelty of our approach lies in the fact that we do not only use one or two usability methods to integrate them into the XP process, in fact we selected many HCI instruments to enhance the existing XP process. This multi-instrument approach was developed to solve all the problems as introduced in Section 1.3.1. 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 usability [145][68][69]. The HCI instruments we relied on are as follows:

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

The integration of these HCI instruments into the XP process is shown in Figure 5.1. The following sections will provide a short overview of the single methods.

Approach to User-Centered Design

User-interface design plays an important role in the acceptance of a web based application by 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 insights into the functional and cognitive requirements of real users. 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 5.2 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, then comes to mock-up designing and at the end to the actual implementation. The process is executed as follows:

- Different feature-related user stories of the application are created by the customer along with partners.

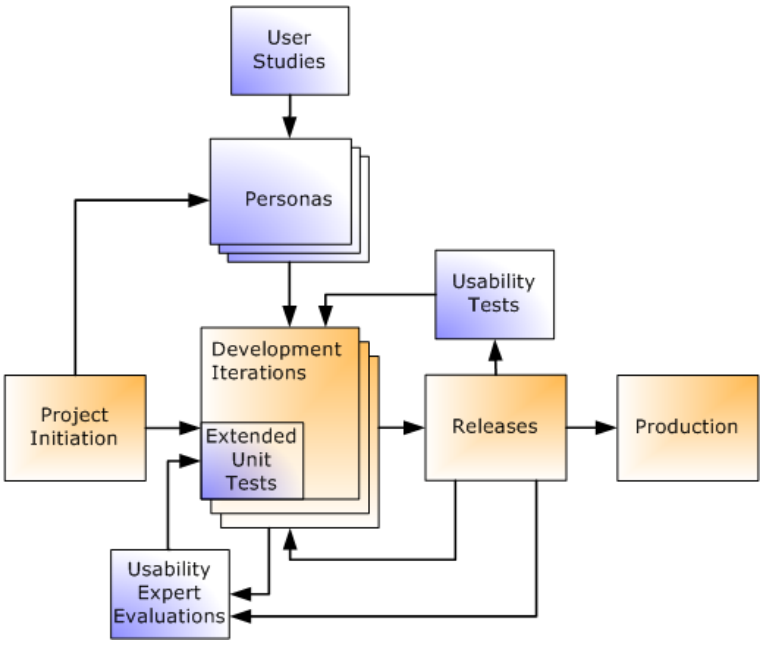


Figure 5.1: The modified agile development process with usability instruments included (Extreme Personas, User Studies, Usability Tests, Usability Expert Evaluations and Extended Unit Tests). End-users are integrated in two different ways into this process: on the one hand user studies inform the development and extension of the personas (which indirectly provides input to the developers); on the other hand usability tests (as a part of the usability evaluations) directly inform the development [145][68].

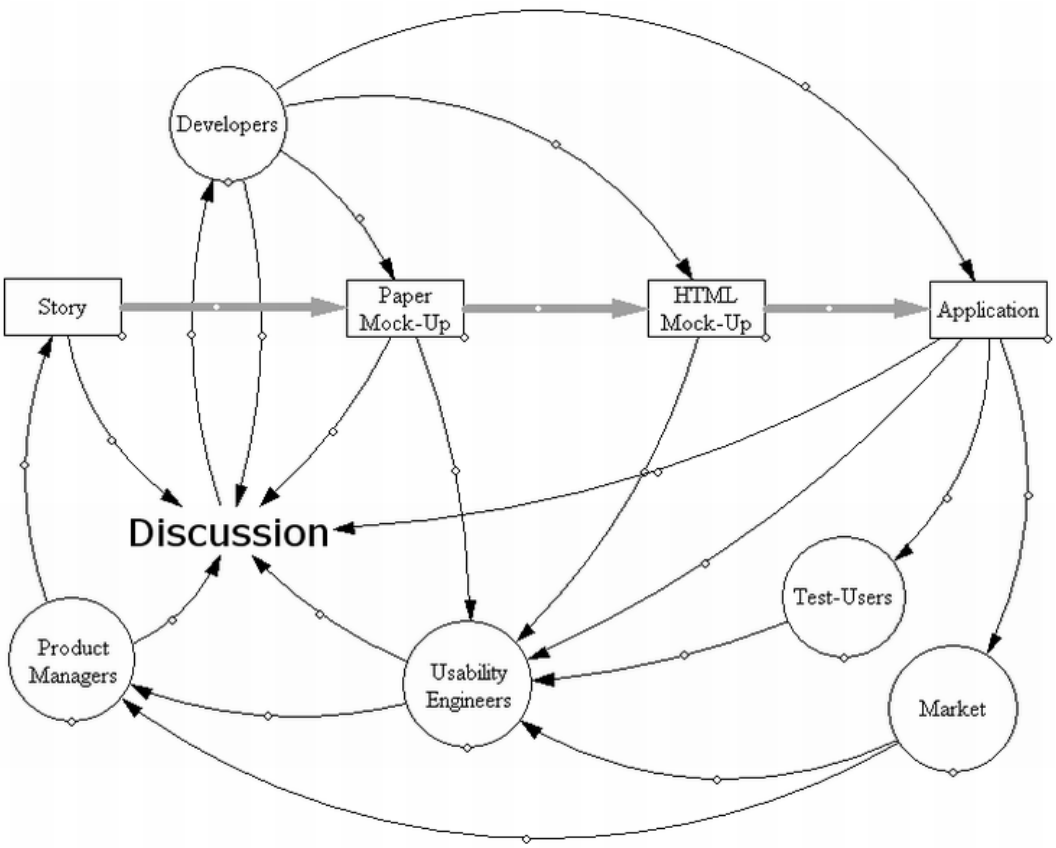


Figure 5.2: Iterative UI design workflow [65].

- 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 of the mock-ups best suits his needs or suggests modifications to the mock-ups.
- A final mock-up is derived according to customer's wishes which then serves as the basis for the actual implementation.
- Once an implementation mock-up of a feature or group of related features is finished, the usability engineers are asked to give feedback on it.
- After incorporating the feedback given by the usability engineers into the application, end-user tests are conducted on the application by the usability engineering team.

- 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 (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 user and the specifications derived from actual usage.

5.1.2 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." [33, 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 [33]. Since the invention of personas, many scientists but also big 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 [145][69]. 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 [33].

They describe the target user - his wishes, desires and application-specific aspects. Furthermore, they show the nature and scope of the design problem [113]. 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 [113]. 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 [33][113].

The development of personas is usually based on a very detailed design process starting with the gathering of background-marketing information, the design of personas skeletons, their introduction, and a marketing campaign [113]. Personas are supposed to be applied not only once in a project, but throughout the entire duration of a project.

The final personas do not only exist on paper. Within a project they come alive through different scenarios and use cases that are built around the personas [113]. Therefore, they can be seen as real people with lives and - if applied right - they will even feel real to the project members.

The advantages of personas are at hand: As mentioned above, they allow to unify the picture of the target user for the project team, which allows for a more fluent communication [113]. Additionally, personas make use of the "emotional mind" [128] of people which leads to a better focus on user-centered thinking within a project. "The user" as abstract term is eliminated from the project and is replaced by people with names and faces, which also saves time because of shortened debates and so on. 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 [33]. Furthermore, personas can also be used as an evaluation tool, such as walkthroughs [113]. As an archetypical figure, personas can guide decisions about product features, navigation, interactions, and even visual design (among other factors) [104].

In the agile development process, personas can be integrated as so-called "Extreme Personas" [145], an approach to personas that starts with the same activities like the classical persona method: preliminary user groups are defined and personas are modeled for them afterwards.

In addition, the knowledge gathered in user studies is incorporated and the personas will be refactored, when the 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 the Extreme Personas yield as

reference representation of the on-site customer.

5.1.3 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 refactoring the personas is obtained; on the other hand direct input for the user stories can be derived [145][69].

In the agile process user studies were employed in two different ways [69]:

- Laddering interviews and
- Field trials

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

Laddering Interviews

Laddering interviews are techniques that are mostly applied in the field of marketing and psychology. Nevertheless, recently it has also been applied to investigate user experience [112]. In a structured interview between two persons, the connections between attributes and their consequences are investigated from the interviewee's point of view [117].

The duration of a laddering interview can be - depending on the content - between forty-five minutes and two hours. The 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.

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.

5.1.4 Usability Tests

Usability tests are empirical studies that involve real users testing an application [69]. They can be considered as the most fundamental usability evaluation method [103]. 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) [145].

5.1.5 Expert-based Usability Evaluations

Expert-based usability evaluations are reviews conducted by experts [69], either in the area of usability, in the application area of the particular system, or both. The following two expert-based usability inspection methods are the most renowned:

- Heuristic Evaluation
- Cognitive Walkthrough

Heuristic Evaluation

Heuristic Evaluation has gained in popularity with Molich and Nielsen's introduction of ten heuristics to a wider audience [55][101]. The original heuristics have later been improved and adapted to different areas [103] and are still frequently employed to evaluate different kinds of systems.

This evaluation method is considered an efficient analytical and low-cost usability 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 usability 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 usability 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.

Cognitive Walkthrough

A cognitive walkthrough is an analytical usability inspection method which was introduced by Wharton, Rieman, Lewis and Polson [143]. The main goal of a cognitive walkthrough is to measure the learnability of a system by detecting usability 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 [55]. 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.

Application in Agile Environments

In projects involving XP, expert-based usability 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 [145][69]. 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.

5.1.6 Extended Unit Tests

Extended unit tests originate from automated usability evaluation (AUE) [69]. The idea of automated usability evaluation is not new. Basic research goes back to the early nineties [15][49]. In the year 2000 the state of AUE is still described as "quite unexplored" [78].

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 [14][60]. Here the generated data helps identify paths and execution time in order to detect problems.

Another example for AUE is the NIST-Web-Suite[1] that allows for an automatic code-based analysis of websites according to 12 design guidelines. The

[1] http://zing.ncsl.nist.gov/WebTools/

WAUTR-project[2] (Automatic Usability Testing enviRonment) is a first attempt to support usability 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 [6] or specific aspects (e.g., gaze[3]) is one possibility to solve this issue. Nevertheless, also this approach requires user-data.

When there aren't any users 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.) [78][79].

This approach is currently mainly used in the web area. WebTango is one of the best known tools in this field of application [77]. 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 Maysoon Abdulkhair [6], who has 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 [107] 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 [145]. These tools would then be able to continuously check for usability 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[4], 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 [108] and mobile devices [142], 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.

[2] http://wauter.weeweb.com.au/

[3] http://www.goodgaze.com/ggx/, http://www.feng-gui.com/

[4] http://www.usablenet.com/

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 to testing. Therefore, the unit tests are extended by tests, which go beyond single units, and test complete interaction flows [145][69].

5.2 HCI Instruments: Lessons Learned

We use the previously described process since summer 2007 in our project for more than two years. The final usability tests and the field trial show that the process is able to really enhance usability of XP-style developed applications.

The tight coupling of different expertise has led to high motivation among project members. Developers gain insight into the subtleties of UCD, HCI experts learn to understand the origin of usability problems. Especially the diverse technical testing frameworks demand technically aware HCI experts. Depending on the used frameworks, the knowledge needed varies. In practice, this could become a problem when the chosen framework is complex and little time for learning is available [145][68][69].

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 usability-awareness and on the mindset present in the project - not on the software development model.

Furthermore, we found out 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 [69].

Also from a customer point of view the communication with a usability 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 usability engineer actually sees them rather than getting them described. The response times of these methods are usually short enough since for usability input in the story-writing process it is sufficient to get results within 3 - 4 days. For quick feedback on usability issues during development or for "urgent" re-planning usability engineers should be readily available for a quick advice via cell-phone or chat [69]. However, it should be mentioned that we had good experiences with physical visits. This allows the usability 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 provided 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 pairs with the customer in order to create simple and exact 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 usability engineer is also included early into the story creation-process. The advantages are timesaving, more motivation (less chances of rework due to preclarified usability issues), and gaining better realization of needed usability input early in the development [69].

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. Until now we did not experience these problems we assumed would come according to the literature.

5.2.1 Review of the Used Instruments

Over two years of 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:

5.2.2 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 [69]. 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 usability engineers. Second, the development team didn't give much credit to the two personas which were provided [69].

From the point-of-view of the development team and the customer-on-site the main cause for this 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 [69].

What can be learned out of this?

First, Personas should be introduced like new team members: You wouldn't ignore them. The introduction of personas that will accompany the whole team during the development cycle and beyond is as important as the introduction of team members. 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 doesn't matter for whom they develop. The design is supposed to be the design-departments decision, the scope of implementing which feature is part of the customer's work, and over-all everything is pre- and post-tested by the Usability 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 from the part of the customers, the opinion is strong 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.

5.2.3 Expert Evaluations

During our project, it took some time until the customers embraced the possibility of asking the advice of the usability engineer in advance. This might have been caused by the increasing trust in the usability engineer over time. After a while the customers have reportedly valued the usability engineer's input higher. Another reason was the improved planning method which allowed to prepare stories early enough to have time for usability 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, smaller and faster evaluations on story level are needed. If the evaluation result comes after the story completion or iteration, 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 version of the usual expert evaluation process is more practical.

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 usability engineer for feedback [69].

Feedback from the usability-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 usability side would cost too much, it is advised to call the usability-engineer and have a short (video) conference during planning or whenever this occurs. Far less usability-fixes are the results of this practice [69].

Thus, it is our advice to involve available usability engineers into the planning process as early as possible. Use his time and input only when it is intended to implement the results 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 usability input will be realized.

5.2.4 Usability Tests

Usability tests are carried out to evaluate the running prototype. One of the usability studies was executed in January 2008 with 10 respondents using a mobile phone. The classical task-based usability test method was used [121]. Each respondent was asked to execute 5 different tasks. Tasks were carried out on a Nokia N95 mobile phone. To gather general feedback and general opinions, two interviews were carried out: One before and one after the task session (pre- and post-interview). Each task was accompanied by task specific post-questionnaires. Interview sessions lasted about 1 hour. For the tests the device's standard browser as well as opera-mini 3.0 were used (the first is incorporating a web-like mouse pointer, the latter a link marker to navigate through the interface).

After the test, respondents had to judge three different visual design paper-prototypes. We used the AttrakDiff questionnaire [51] to capture the attitudes of the users towards the application in terms of graphical design, enjoyment, and aesthetics. After the tasks, the AttrakDiff questionnaire was filled-out.

The results of the tests suggest two main improvements:

Improvements of Layout and Design:

Main improvements should be made concerning the visual appearance of the site:

a) The actual site, menu, and navigation layout is not optimal. Through the use of the color blue as text color and background color at the same time, equal text sizes throughout the interface and different alignments, the site's hierarchy is not visible for users.

b) The actual layout does not incorporate visually attractive design elements and is rated as pragmatic but monotone with a lack of stimulating elements (Attrakdiff questionnaire).

Figure 5.3 shows the prototype of the home page.

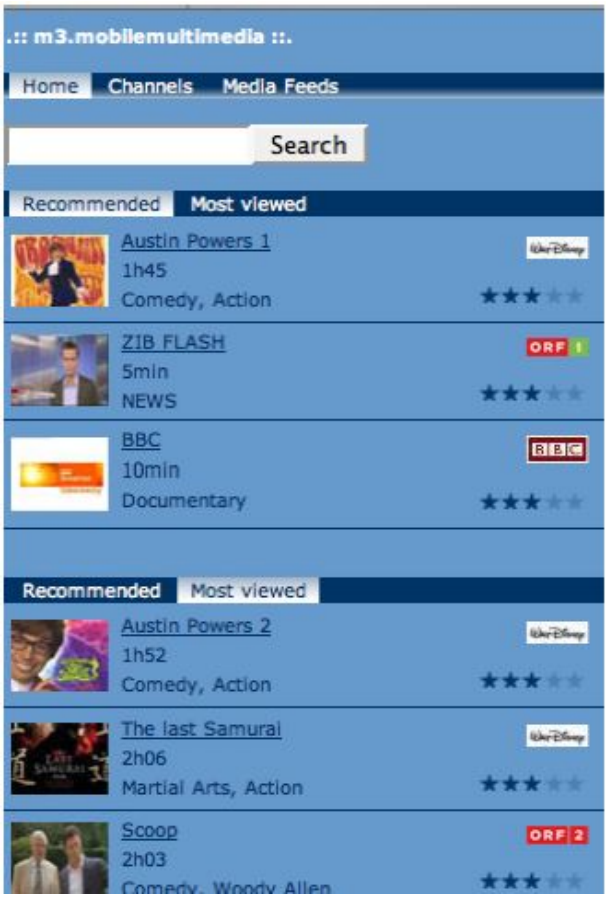


Figure 5.3: The prototype of the home page

Improvements of the Prototype's Usability:

On the "Channel" web page, a web-like calendar function to select dates should be integrated (the actual function will not be usable for greater amounts of data). All navigation menu elements should be separated from content menu elements ("Home" vs. "Watch"). Furthermore, interactive elements ("Rate", "Comment" etc.) should be placed on a separate page instead of on the bottom of a description page. Figure 5.4 shows the recommended prototype of the Channel page showing the calender. Figure 5.5 shows the Menu entries without any visual separation.

For the further development of the prototype the sub-site "Media Feeds" should be separated into two categories introducing the sites "create Media

Figure 5.4: The prototype of the Channel page showing the calender

Feed" and "watch Media Feed". Special attention should be given to feedback mechanisms which at the moment do not support the user (feedback of search queries, display of media feed search results).

From the mock-ups of three different designs, the AttrakDiff results suggest that a yellow design was most liked by the respondents. It was also suggested that the blue design may be used in the future but the following improvements should be made.

- Accentuate contrast on whole site.
- Avoid light blue text on darker blue backgrounds.
- Introduce visually attractive design elements that increase the attractiveness of the site.
- Eliminate monotony by introducing more colors.

Two of the developers also observed the usability study session which gave them a chance to realize the impressions of actual end-users and their feelings as well as actual usage. This helped in guiding the development according to the wishes of end-users.

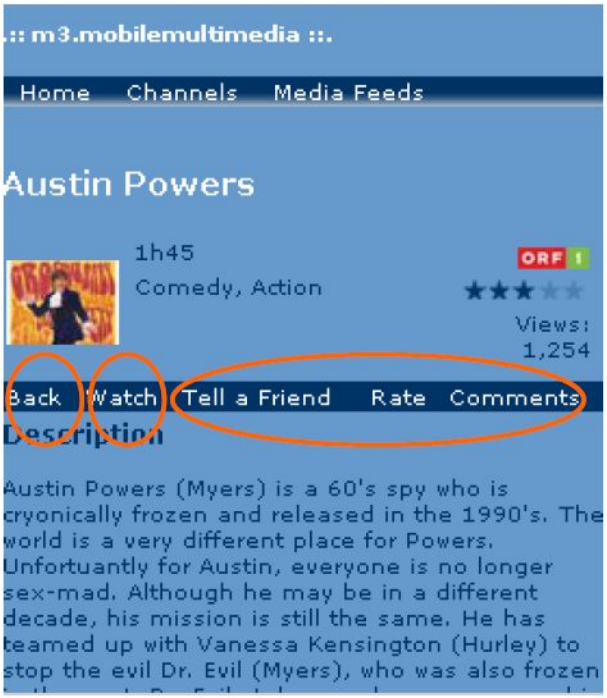


Figure 5.5: The menu entries without any visual separation

A Task Example

Here a task example is presented. The task is: "Find the detailed description of a given movie, write a comment and rate it".

Facts on Task:

The task was completed without any greater difficulties by all respondents. On the "Home" page and on the "Channel" page respondents used the heading to find the detailed description and the video's thumbnail to watch the video. Respondents did not encounter much problems on the "Clip Detail" page. The prominent position of the links "Comments", "Rate" and "Tell a friend" – Figure 5.5 – on top of the description page helped respondents to understand which possibilities are offered. On the "Clip Detail" page there are two interaction paradigms that were both understood: Clicking on the link "Tell a fried" opens a new page. This didn't cause any problems for users. The functions "Rate" and "Comment" are placed at the bottom of a "Clip Detail" page and users had to scroll down or use a link to jump down. In reference to both described

paradigms, user comments indicate that the longer the list of comments is, the more uncomfortable the site is to browse. Further, the task uncovered that on the mobile interface respondents did not recognize that they were scrolling down the page when using the anchor-links "Comment" and "Rate". To get back to the top of the site they pushed the "back" button. This did confuse some of the respondents as they jumped back to "Home" although their intention was to get to the top of the "Clip Detail" page. Of course this depends on how the browser implements the "back" functionality.

A solution that incorporates interactive functions ("Rate", "Comments", "Tell a Friend") on a separate page is recommended. Furthermore:

- Back Button: A dedicated back button should be integrated on top of the page. This is where basic navigation elements are expected.
- Watch Button: A "watch" button should be designed and integrated consistently. An additional watch button – if necessary – should be placed on a particular spot on the site and not be integrated in the navigation menu. The "Watch" button should be visually highlighted.
- Tell a friend, Rate and Comments: These elements describe "interactive functions" on the site and therefore should be kept together and aligned to the left side of the page.

Figure 5.6 shows the recommended menu layout and arrangement.

Respondents' Feedback/Comments:

- All respondents indicated that in their opinion the "Clip Detail" page provides a good overview.
- The design of the "Comments" and "Rating" section is good and intuitive. Too many comments on one page should be avoided as the page would get too long (1 respondent).
- Comments should be ordered in chronological sequence, beginning with the most recent entry (1 respondent).
- The space on top of the "Clip Detail" page (heading) should be used in a better way. This would provide more space for description texts (1 respondent).

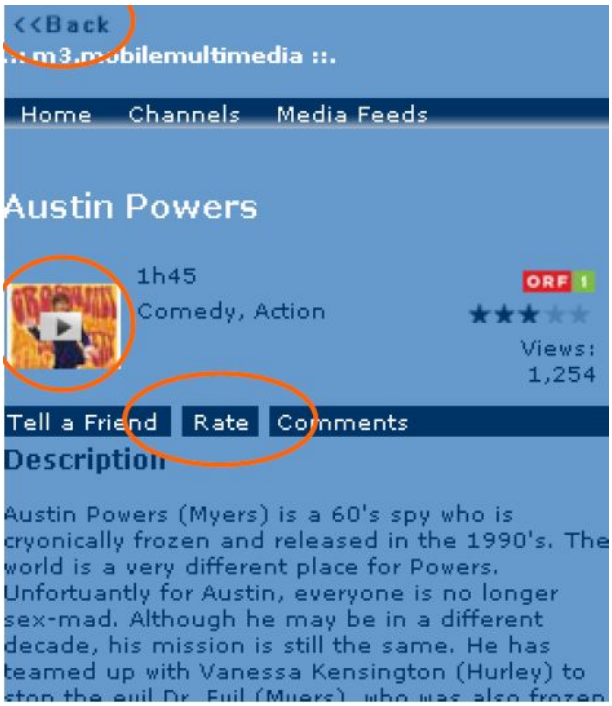


Figure 5.6: Improvements of menu layout and arrangement

- It should be possible to select which information is sent to another person via the "Tell a friend" function (the video's description, the video itself, etc.). Radio buttons should be used to specify one out of different possibilities (1 respondent).

Figure 5.7 shows the space on top of the "Clip Detail" page which should be used more efficiently.

What can be learned out of this?

We noticed that the usability tests had impact beyond the expected (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 [68][69]. We saw that developers who attended the usability tests got more biased towards user-centered thinking than the others [68]. 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.

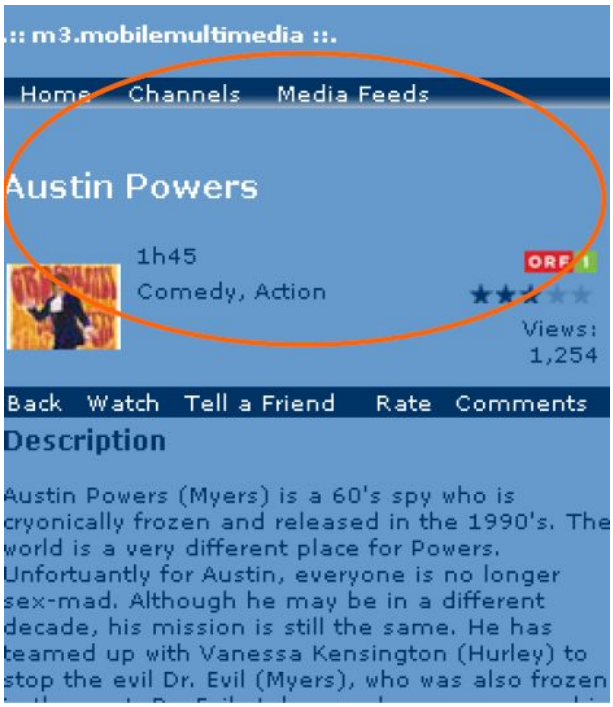


Figure 5.7: Use the space on top of the "Clip Detail" more efficiently

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 usability 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 already obsolete [69].

Therefore, we would recommend smaller tests after every few iterations of the application (better for every iteration). To keep the costs feasible and since 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 [69].

5.2.5 User Studies

We used user studies in the form of laddering interviews and field studies. In autumn 2007 the laddering interviews were conducted; the results were

published in [90]. 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, and focus group [69]. We also logged the important actions of the users. The log files results are presented in Chapter 6. Other trial results will be presented in future publications.

5.2.6 Extended Unit Tests

We implemented the automated usability evaluations in the form of extended unit tests. The tests were written prior to the implementation of any user interface related functionality. The tests were also written prior to fixing any user interface related usability issue. The tests have also been included in the continuous integration and the nightly build process. This resulted in high transparency regarding the usability of the application.

5.2.7 Lightweight Prototypes

Besides the HCI instruments mentioned above, we also make use of two different types of mock-ups; low fidelity paper mock-ups and high fidelity mock-ups and get them evaluated by the customer. As both the developers and the customer have been increasingly gaining knowledge about usability engineering, evaluating paper prototypes with the customer is good on one hand but on the other hand the customer has now become an expert user instead of a casual or novice user. So there is always a chance to ignore the actual needs of real casual end-users. We have mitigated this risk by having more ad hoc input from the usability engineer and suggested to conduct a formal usability test with at least 10 end-users after every release, i.e., quarterly. For special "ad hoc" questions instant messaging is used to gather HCI feedback within a short time frame from the usability engineer regarding stories or mock-ups. It would have been more beneficial if the usability engineer would have been present on-site to quickly give his feedback, and instead of developers he should do the prototyping not only with the customer but also with at least a few end-users [69].

5.2.8 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 in the following.

Especially from the customer's point of view it is recommended to learn to cope with possible daily occurring misunderstandings in the use of business and technical-tongue. During the planning process clarification is particularly important.

Furthermore, the usability expertise should be integrated very early in the planning phase and also during the story-writing process. Invite your usability 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.

Overall Tips & Tricks

From a customer's perspective it is important to be careful about the extent of how much you apply the YAGNI principle. It can cost quite some time and money to overdue YAGNI when, e.g., not implementing a technology whose usage is yet to come, when it is "sure" that this will be the case. Nevertheless, try to use YAGNI as much as possible and be very clear about what is "necessary" for the CURRENT task. As this kind of thinking has to be trained and is against the usual way developers and customers work, there is a big tendency to think too complicated and to anticipate all possible future directions.

Prepare for planning: the more can be clarified in advance the easier the planning goes. Sign up for stories: one pair should clearly sign up for a story, so that it is always known by everyone who is working on what story at the moment.

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 usability 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).

Tips & Tricks for Extreme Programming Projects

Boiled down to a short list we summarize the actual state of experience in this list of tips and tricks:

- 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 usability findings in form of user-stories. This saves an additional step and makes them immediately usable during the planning game [69].
- Story writing: a developer should support the HCI engineer when writing user-stories [69].
- Proper Customer and HCI engineer coordination is necessary for the inclusion of the usability process in the development [69].
- An experienced on-site XP customer can fill in the technical gap between HCI engineers and the developers.
- A short usability workshop should be held at the start of the project or before the release in which the usability testing phase is being started. It serves as a good platform for all of the XP team members (managers, customers, and developers) to understand the process and importance of usability testing. It also serves as a good starting point for both 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 usability aspects in his user stories.

5.3 XP Practices: Lessons Learned

5.3.1 Testing Issues

A big issue in mobile user-interface design practice is that current approaches are not sufficient for mobile phones [132]. For designing any software, use of good UCD practices ensure that the product is accepted by the users [85] thus supporting 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. 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 on the small device with its limited user input capabilities [85]. Therefore, in our case the application is tested on mobile phones and not on any web based simulator for understanding the interaction issues concerning the use of mobile phone interface [85].

5.3.2 Communication and Collaboration

Communication between stakeholders is an important characteristic of software development. Communication and collaboration between customers, business partners, developers, and other stakeholders enhance overall team efficiency

[94]. The value of communication is expressed by the XP practices of pair programming, metaphor, informative workspace, simple design, on-site customer, the planning game, and coding standards. [52]. Other factors in communication are the use of whiteboards, positioning and sharing of desk facilities to facilitate pair programming, stand–up meetings, developers buying-in to the concepts of the rules and practices of XP, and collective code ownership [43]. We sit side by side in a spacious room having enough space for private workplaces as well as for three separate pairing stations. This seating arrangement has promoted effective interaction in the team and has helped in resolving technical issues on the spot [87]. The teams' XP room is equipped with six whiteboards which are used to record the XP stories agreed at release and iteration planning meetings. Story cards are physically stuck to the whiteboards in prioritized order with adjacent notes written on the board. Various graphs showing architecture and velocity of the project are also drawn on the whiteboards. By looking at the whiteboards, anyone can see the current status of the project.

Email, phone calls, and video conferencing are the tools used in routine communication with the usability engineers and with other partners. Personal visits to and by project partners are also made by the product manager and by other team members whenever necessary.

5.3.3 The Planning Game

We hold two types of planning meetings: release based meetings and iteration based meetings. A release lasts for three months whereby within a release, an iteration lasts for two weeks. Project partners attend release meetings and through discussions identify and define user requirements called user stories in XP [62]. "The parallel with the UCD approach is obvious here: an understanding and appreciation of the users and their requirements" [94]. These stories are written down on story cards and are prioritized by the project partners. Developers then estimate the time required for implementing the stories.

At the beginning of each iteration, an iteration meeting is held and is attended only by the team members including the product manager. The product manager selects and prioritizes stories which fit in the current iteration depending upon the available velocity. Then, developers break down the stories into detailed tasks and estimate them. Finally, the product manager defines the acceptance criteria for each story and task.

Before and after implementing these stories, continuous feedback is obtained from the usability engineer. Then, these stories are modified according to the

feedback of end-users and the usability engineer. Once "again this is a common step with UCD approaches; an understanding of the user goal and the tasks to achieve that goal. Addressing a requirement in terms of the user and their goals focuses development upon what is needed" [94].

The figures 5.8 and 5.9 show story cards of release and iteration plannings stuck on whiteboards.

Figure 5.8: Selected story cards on the Release-Board (Release Planning)

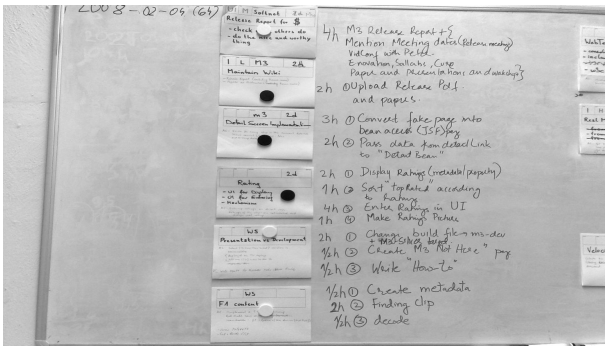


Figure 5.9: Selected story cards on the Iteration-Board (Iteration Planning)

5.3.4 Pair programming

This practice has helped us in spreading and sharing the project-specific knowledge and improving the technical skills of the developers. We also applied the practice of working in pairs with the product manager [62]. The product manager pairs with a developer for writing customer-acceptance tests, thus exposing

the customer to the process and the internal status of the application which helps in better understanding and implementing the end-users' requirements. This also has enhanced the enthusiasm of the team members to work in a collective and collaborative team environment.

5.3.5 On-site Customer

In UCD, "all activities are focused on providing business value through ensuring a useful, usable and engaging product. The customer is not only defined as the project stakeholder, but the end user as well" [94]. The Manifesto for Agile Software Development [91] does not clearly demand end-users as customers. In our process, the product manager plays the role of an "on-site customer" and communicates with the various stakeholders. The end-users are indirectly involved by the usability engineer who provides the HCI instruments.

5.3.6 Small Releases

We aim to release a working version of our application to the project partners on a regular basis. In the early stages of the project, the duration of one release cycle was set to one month, enabling us to quickly get feedback on our work from the partners in order to sharpen our vision of the project goal. As the project took shape, the release size was gradually increased to two and finally to three months. For now we are satisfied with three-month release cycles, which complies with the quarterly planned business targets of the partners.

For tracking short-term progress, releases are further divided into iterations. Initially, we used a one-week iteration duration but later shifted to two weeks in order to reduce the administrative overhead added by the iteration planning meeting.

5.3.7 Sit Together

The team members including the product manager sit in one large room at their private workspaces. There are three separate pairing stations in the same room. Due to sitting at the same location, the face–to–face communication has resolved many difficulties which arose within the project, team and the process.

5.3.8 Collective Ownership

A Subversion repository is used for managing the single code base. The code is shared by all developers, so whenever a chance for code improvement is identified and there is enough time at hand, the required actions are performed on the spot. The changes are communicated in the stand-up meetings, during pair programming, and sometimes through a short ad-hoc discussion involving all developers. One basis for a successful application of collective ownership is the strict adherence to coding standards.

5.3.9 40-hour Week

The purpose of "40-hour week" practice is that developers should not work during overtime otherwise tired developers can make more mistakes during coding [52]. We strictly follow this practice.

5.3.10 Test-first Programming

As all the team members were new to XP, it was difficult to follow the XP style of writing the failing automated test before any code [18] [19].

Figure 4 shows a graph comparing lines of executable code, lines of test code and test coverage. For this the data was collected using Emma, a Java code coverage tool [2] and LinesOfCodeWichtel [4], and is based on the work performed during the second release of the project. The low amount of test code and coverage show that the practice of testing is not well exercised by the team. For the subsequent releases, the implementation of this practice has improved. Whenever possible, tests are being written beforehand.

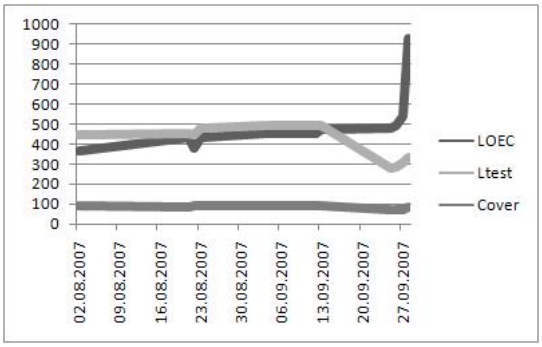


Figure 5.10: Executable code versus test code and coverage

5.3.11 Reflection

In order to measure the performance of the team and also resolving human issues, a retrospective review meeting on the process is held after every iteration and release. This retrospective meeting is called "reflection meeting". It has helped us a lot to find out the reasons for difficulties faced during the process and to find out their remedies. The common decisions that we take after these meetings are noted down and followed by all team members. Almost all XP values – communication, simplicity, feedback, courage and respect – and XP practices adhere to human aspects [18] [19]. The benefits of sitting together, face-to-face communication, feedback, stand-up meetings, the planning meetings, pair programming and reflection meetings have contributed to improve not only our process but also to increase the overall morale of the team.

To review the development process, we collect empirical data from various sources describing our performance of each applied XP practice.

For a qualitative analysis, we perform the Shodan 2.0 survey [86] on a regular basis (i.e. at the end of each release). Additionally, we use quantitative data generated by the XP tracker tool "xplanner" [5] and different code analysis tools [2] [4].

The data gathered using Shodan 2.0 Input Metric Survey shown in Table 5.1 gives an overview about the methodology and the extent to which a given XP practice is applied. As there was an explicit effort to apply these practices, low percentages indicate that either the team was not fully content with the practice or the practice needs to be tailored for our project. For example, the team members perceived that pair programming was practiced for almost every development task, while metaphor was given the lowest rating because of the unfamiliarity of the team members with the project. These conclusions are also supported through iteration and release reflection notes and from key discussion points raised in stand-up meetings.

5.4 Summary & Future Research Directions

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

The future research direction is presented in Section 7.1.2.

Testing metrics	%
Test First Design	44
Automated Unit Tests	68
Customer Acceptance Tests	22
Planning metrics	%
Stand-up meetings	92
Short Releases	86
Customer Access / On-site Customer	48
Planning Game	96
Coding metrics	%
Pair Programming	98
Refactoring	66
Simple Design	76
Collective Ownership	86
Continuous Integration	100
Coding Standards	84
Sustainable Pace	82
Metaphor	46

Table 5.1: Subjective metric (Shodan 2.0 Input Metric Survey)

Chapter 6

Analyzing Real Mobile Web Usage of a Multimedia Streaming Application Through Log Files

Web use on mobile phones is increasing day by day. Various web applications and services are emerging on mobile phones. One such application is consuming multimedia – Audio and Video (AV) – on mobile devices. We have been working on an application that enables a user to perform content-based search for AV content in large digital archives and play the streamed content on a mobile phone. The search is not only possible in meta data but the application is intelligent enough that it also searches in spoken words and shows the context of spoken words in the clip so that user can immediately watch the particular part of the clip where these words were spoken. The chapter describes the application features. The chapter also describes the preliminary results of a field trial study with 150 users from the perspectives of the real usage of the application and its web features. Main reliance was on quantitative methods used during the study for the analysis in the form of log data of the application. The main goal of the field trial study was to examine the acceptability of the mobile multimedia streaming application by consumers

6.1 Introduction

Mobile phones have become full-featured mobile computers [65]. They are one of the most commonly carried personal, readily accessible technology. Besides voice calls, they have capabilities of digital cameras, PDAs, browsers, and audio-video players [111]. Web use on mobile phones is increasing day by day. Various web applications and services are emerging on mobile phones. According to a research survey study by Ipsos Insight, "Mobile phones could soon rival the PC as World's dominant Internet platform" [74]. This has already happened in Japan where the study by comScore estimated that mobile phone web users (53.6 million) are nearly equal to PC based Internet users (53.7 million) [31]. A report by "Gartner estimates that by 2010, more than 50% of cellular subscribers in the U.S. and Western Europe will access the web on a mobile device at least once a week" [42] [120]. Google is taking lead in this direction and provides many services on mobile phone, e.g., Google Mobile Search, Gmail, Google maps for mobile, mobile YouTube, etc. Even though the number of mobile web users is growing, the satisfaction level of the users remains low; only 12.6% of the comScore study participants were either very satisfied or somewhat satisfied [31]. User Experience (UX) of these applications and services for mobile phones provides unique challenges for the users and for the service providers. There is a need to study the factors affecting the UX of a product or a service [138].

It is stated that multimedia -AV- consumption is going to be one of the popular applications for mobile devices [65] [36]. A huge amount of precious AV content is kept unused in archives of content providers [65]. We have been working on an application that enables a user to perform content-based search for AV content in large digital archives and play the streamed content on a mobile phone. The applied process to application development focuses on the adoption of agile software development methodologies and user-centered design [65].

The application features are already described in Section 2.4.1. The chapter also describes the preliminary results of a field trial study with 150 users from the perspectives of the real usage of the application and its web features. We mainly relied on quantitative methods used during the study for the analysis in the form of log data of the web application as well as the streaming server. Using log file analysis has the advantage to investigate users' natural behavior and actual use of an application or a service. Various studies are available for log analysis mainly for web applications but recently also for mobile phones [35]

[80]. For further supporting the analysis, qualitative methods were also used in the form of diary studies, contextual interviews, questionnaires, usability tests, and focus group. In this chapter, the preliminary results of log file analysis of the application and its web features are described, i.e., which application features were mostly used; which keywords were mostly searched; how frequently users rated the clips; how frequently users made comments about the clips, etc. The past work ranges from covering satisfaction of the needs of users by focusing on usability to the integration of agile software development method with User-Centered Design in all steps of the development process [145] [65] [68] [66] [69].

The following Section 6.2 describes the field trial study and results of the log file analysis. Finally, conclusion is presented.

6.2 The Field Trial Study

The field trial study took-place with 150 real end-users spread throughout Austria. The study started in December 2008 and ended in May 2009. The field trial study is multi-purpose as various stakeholders of the project are interested in getting knowledge about their own business and research needs. The main goal is to examine the acceptance of the multimedia streaming application by consumers.

6.2.1 Selection of Users

For the field trial study 150 users were chosen in the age group between 18 and 65. These users were chosen by the mobile network operator who already had all data about these users, i.e., users' socio-demographic data as well as data about their usage of the mobile Web. Additionally, it was required that users should have some experience in using mobile multimedia. These users were provided free unlimited access to our application during the field trial study. The application was used on Nokia 6210 Navigator and Nokia N95 mobile phones. 100 users were given the Nokia 6210 Navigator as a compensation of the field trial study. Additionally, 50 users were asked to use their own mobile phones. All 150 users used their own SIM cards. The application was not used continuously for six months by all users. Some started using the application in the early phase of the first month and most started in the middle phase while few started in the ending phase. From them 60 users were selected for writing their daily usage diaries; 60 users were asked to answer the contextual

questionnaires; and 30 users were left totally undisturbed. 16 users took part in a laboratory usability test. Additionally, 16 users were selected for a focus group.

6.2.2 Study Results

The log data contained a unique user ID (anonymous), timestamp, URL, clip ID, keyword searched, rating, comment, recommendation, and many other data fields.

According to the log file analysis, other than home page the most accessed page was the Channel page (26.04%), followed by the Category page (24.94%). The SearchResults page was accessed in 18.98% of the accesses for displaying search results.

On average per day per user 10.3 minutes of clips were watched which shows that the users liked the application and watched the clips on even a small screen of mobile phones. The most searched keywords are "Simpsons" 10.56%, followed by "charlie" 2.79%, followed by "mein cooler" 2.64%, followed by "scrubs" 2.54%, and "zib" 2.29%. First keyword is related to a cartoon series, the next three keywords concern sitcoms while the last keyword is about news category, respectively. The average number of search queries performed by each user per week is 1.637, which shows that users preferred browsing the application especially the Channel page, rather than searching. This is quite reasonable due to the limitations of mobile phone's small keypad to enter text. Additionally, it is also due to the fact that clips inside the Channel page are ordered by time, e.g., by the broadcast time for TV channels which can be browsed and watched easily. There are 11 categories which are shown here in the order of containing more clips: "Serie (series), Information, Film, Sport, Magazin (magazine), Show, Kinder (children), Misc, Doku (documentary), Kultur (culture), and Neutral". It should be noted that both the application as well as the clips are in German language as the users are native German speakers.

6.3 Conclusion

The field study results by Roto [119] show "that taking the user experience characteristics into account helps creating positive user experiences". Our field trial study relied mainly on quantitative methods used during the study for the analysis in the form of log data of the application. Using log file analysis

has the advantage to investigate users' natural behavior and actual use of the application as well as its features, so that not only the usability but the overall UX of the application can be improved.

The real usage behavior of the application by users show that users browsed the application more rather than using the search feature, although the number of searches is also quite high. This behavior is due to the tiny keypad of mobile phones. The total time spent by users using the application and watching clips is reasonable which shows that users are willing to use the application and watch the clips on even the small screen of a mobile phone. Overall, the results indicate that the users liked the application.

The results could be used to develop the personas for guiding the development process and the team in order to improve the overall user experience of the application and make it more acceptable among consumers. New requirements and fixes will also be captured to further improve the application and its user interface. Along with the log files data we have also focused on qualitative data from users. Other qualitative results of the field trial will be presented in future publications. These results will be compared with log file analysis for further gaining insights to improve the mobile web experience of the application and its users, and making the application more acceptable among the users.

Chapter 7

Conclusion and Future Research Directions

This final chapter concludes this thesis with future research directions.

7.1 General Conclusions

This thesis investigated the relationship between agile development methods and user-centered design methodologies. The main research question of this thesis was: How successfully are these two important methodologies being integrated in industry, and interestingly how one can successfully integrate them. This led to further interesting questions: How the role of a UCD professional is carried out in organizations? What types of HCI techniques are used in teams/projects employing agile methods? To what degree are agile development methods and usability/UCD perceived by practitioners as being integrated and added value to their adopted processes and teams?

In the case study mentioned in this thesis, several HCI techniques were integrated in Extreme Programming agile process for more than two years. The results presented in Chapter 5 show that integrated agile user-centered design process was successfully implemented. The HCI techniques that were integrated into the Extreme Programming agile method were: user studies, extreme personas, usability expert evaluations, usability tests, automated usability evaluations in the form of extended unit tests, and both high and low-fidelity paper prototypes. User studies contained laddering interview techniques by means-end theory and field trial. The tips and techniques for the practitioners, the solutions, and the lessons learned are already presented in Chapter 5.

The mentioned integrated agile UCD process can be successfully introduced in the projects having similar context. Particularly, the use of low-fidelity prototyping, usability expert evaluations, and rapid iterative testing easily fit within the fast moving iterations of agile methods. This was further supported by conducting a worldwide online survey (Chapter 3) and interviews from several agile UCD teams that were analyzed through Grounded Theory (Chapter 4). The results show that the majority of the respondents of survey and interviewing teams have dedicated UCD professionals mostly co-located. Where there is no dedicated UCD professional present in teams then this role is performed by developers having interest in HCI. The top most used HCI techniques are low-fidelity prototyping, conceptual designs, observational studies of users, usability expert evaluations, field studies, personas, rapid iterative testing, and laboratory usability testing. The practitioners perceive that the integration of agile methods with usability/user-centered design has added value to their adopted processes and to their teams; has resulted in the improvement of usability and quality of the products developed; and has increased the satisfaction of end-users of the products developed. The findings give hope and new prospective for better usability and quality of the products developed through integrated agile user-centered design process as well as increase the hope that both communities of UCD professionals and agile practitioners can work even closer to create useful and usable products.

7.1.1 The mobile multimedia streaming application

The mobile multimedia streaming application was successfully developed by employing the integrated agile UCD process. This was evaluated in the field trial, usability tests, and through log file analysis (Chapter 6) showing that the usability and the overall user experience of the developed application has significantly improved.

7.1.2 Future Research Directions

There are several directions for future research. One is automated usability evaluation (AUE) and its further 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 exponentially growing number of custom software products is not

accomplished with 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 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 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.

Additionally, the integrated agile user-centered design process will be implemented in the different context and the nature of the project for its validation and/or getting new experiences about the adopted process.

Bibliography

[1] http://www.ambysoft.com/surveys/success2007.html. last visited: 15 Jan 2008.

[2] Emma: Java code coverage tool. http://emma.sourceforge.net/.

[3] ISO 13407: Human Centred Design Process for Interactive Systems. http://www.ash-consulting.com/ISO13407.pdf. Last visited 1-5-2010.

[4] LinesOfCodeWichtel. www.andreas-berl.de/linesofcodewichtel/en-/index.html.

[5] Xplanner: (xp) project planning and tracking tool. http://www.xplanner.org/. Visited on 04.01.2008.

[6] Maysoon Abdulkhair. *A Multilingual Automated Web Usability Evaluation Agent*. PhD thesis, University of Sheffield, England, 2004.

[7] Pekka Abrahamsson, Juhani Warsta, Mikko T. Siponen, and Jussi Ronkainen. New directions on agile methods: a comparative analysis. In *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*, pages 244–254, Washington, DC, USA, 2003. IEEE Computer Society.

[8] Steve Adolph, Wendy Hall, and Philippe Kruchten. A methodological leg to stand on: lessons learned using grounded theory to study software development. In *CASCON '08: Proceedings of the 2008 conference of the center for advanced studies on collaborative research*, pages 166–178. ACM, 2008.

[9] Scott W. Ambler. Disciplined agile software development: Definition. http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm.

[10] Scott W. Ambler. Agile practices and principles survey results: July 2008, 2008. http://www.ambysoft.com/surveys/practicesPrinciples2008.html.

[11] Scott W. Ambler. *Tailoring Usability into Agile Software Development Projects*. Springer London, 2008.

[12] Scott W. Ambler. Agile practices survey results: July 2009, 2009. http://www.ambysoft.com/surveys/practices2009.html.

[13] John Armitage. Are agile methods good for design? *interactions*, 11(1):14–23, 2004.

[14] Richard Atterer and Albrecht 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*, pages 1347–1350, New York, NY, USA, 2007. ACM.

[15] Sandrine Balbo, Joelle Coutaz, and Daniel Salber. Towards Automatic Evaluation of Multimodal User Interfaces. In *IUI '93: Proceedings of the 1st International Conference on Intelligent User Interfaces*, pages 201–208, New York, NY, USA, 1993. ACM Press.

[16] Jeremy T. Barksdale and D. Scott 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*, pages 4691–4694. ACM, 2010.

[17] Kent Beck. Embracing change with extreme programming. *Computer*, 32(10):70–77, 1999.

[18] Kent Beck. *Extreme Programming Explained: Embrace Change (1st Edition)*. Addison-Wesley Professional, 1999.

[19] Kent Beck and Cynthia Andres. *Extreme Programming Explained : Embrace Change (2nd Edition)*. Addison-Wesley Professional, November 2004.

[20] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley, Boston, 2004.

[21] H. Beyer, K. Holtzblatt, and L. Baker. An agile customer-centered method: Rapid contextual design. In C. Zannier, H. Erdogmus, and

L. Lindstrom, editors, *XP/ Agile Universe*, volume 3134 of *LNCS*, pages 50–59. Springer-Verlang Berlin, 2004.

[22] Stefan Blomkvist. *Towards a Model for Bridging Agile Development and User-Centered Design*. Springer Netherlands, 2005.

[23] D. Broschinsky and L. Baker. Using persona with XP at LANDesk software, an avocent company. In *Agile, 2008. AGILE '08. Conference*, pages 543–548, 2008.

[24] J. Brown, G. Lindgaard, and R. Biddle. Stories, sketches, and lists: Developers and interaction designers interacting through artefacts. In *Agile, 2008. AGILE '08. Conference*, pages 39–50, 2008.

[25] Michael Budwig, Soojin Jeong, and Kuldeep 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*, pages 3075–3084. ACM, 2009.

[26] Bendik Bygstad, Gheorghita Ghinea, and Eivind Brevik. Software development methods and usability: Perspectives from a survey in the software industry in Norway. *Interacting with Computers*, 20(3):375–385, 2008.

[27] Lan Cao and Balasubramaniam Ramesh. Agile requirements engineering practices: An empirical study. *IEEE Softw.*, 25(1):60–67, 2008.

[28] Stephanie Chamberlain, Helen Sharp, and Neil 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*, volume 4044 of *LNCS*, pages 143–153, Heidelberg, Germany, 2006. Springer Verlag.

[29] Tsun Chow and Dac-Buu Cao. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 2008.

[30] Gerry Coleman and Rory O'Connor. Using grounded theory to understand software process improvement: A study of irish software product companies. *Inf. Softw. Technol.*, 49(6):654–667, 2007.

[31] comScore. Mobile web metrix: Japan report. a market research report, short summary available at http://www.comscore.com/press/release.asp?press=1742, September 2007. Last visited 24th April 2009.

[32] Larry L. Constantine and Lucy A. D. Lockwood. Usage-centered software engineering: an agile approach to integrating users, user interfaces, and usability into software engineering practice. In *ICSE '03*, pages 746–747. IEEE Computer Society, 2003.

[33] Alan Cooper. *The Inmates Are Running the Asylum*. Macmillan Publishing Co., Inc., Indianapolis, IN, USA, 1999.

[34] David Dayton and Carol Barnum. The impact of agile on user-centered design: Two surveys tell the story. *Technical Communication*, 56(3), August 2009.

[35] Rachel Demumieux and Patrick Losquin. Gather customer's real usage on mobile phones. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 267–270, New York, NY, USA, 2005. ACM.

[36] Jen-Wen Ding, Chin-Tsai Lin, and Kai-Hsiang Huang. ARS: An adaptive reception scheme for handheld devices supporting mobile video streaming services. In *International Conference on Consumer Electronics. ICCE '06*, volume 1, pages 141– 142, 2006.

[37] T. Dyba and T. Dingsoyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10):833–859, August 2008.

[38] Melissa Federoff, Catherine Courage, and Craig Villamor. Agile success: A user experience case study. UPA 2009 International Conference. Usability Professional's Association, 2009.

[39] Jennifer Ferreira, James Noble, and Robert Biddle. Agile development iterations and UI design. In *Agile 2007*, pages 50–58. IEEE Computer Society, 2007.

[40] Jennifer Ferreira, James Noble, and Robert 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*, volume 4536 NCS, pages 9–16, Heidelberg, D-69121, Germany, 2007. Springer Verlag.

[41] 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*, pages 63–72, 2008.

[42] Gartner. Mobile Web Trends 2007 to 2011. A market research report published on (18th June 2007).

[43] Robert Gittins and Sian Hope. A study of human solutions in extreme programming. In *In G. Kadoda (Ed). Proc. PPIG 13*, pages 41–51, 2001.

[44] Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine, 1967.

[45] 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.

[46] Jan Gulliksen, Inger Boivie, Jenny Persson, Anders Hektor, and Lena Herulf. Making a difference: a survey of the usability profession in Sweden. In *NordiCHI '04: Proceedings of the third Nordic conference on Human-computer interaction*. ACM, 2004.

[47] Jan Gulliksen, Bengt Göransson, Inger Boivie, Stefan Blomkvist, Jenny Persson, and Äsa Cajander. Key principles for user-centred systems design. *Behaviour & Information Technology, Special Section on Designing IT for Healthy Work.*, Vol. 22 No. 6:397–409, 2003.

[48] Rich Gunther, Jeff Janis, and Scott Butler. The UCD decision matrix: How, when, and where to sell User-Centered Design into the development cycle. 2001.

[49] Monty L. Hammontree, Jeffrey J. Hendrickson, and Billy 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*, pages 431–432, New York, NY, USA, 1992. ACM.

[50] Bo Hansen Hansen and Karlheinz Kautz. Grounded theory applied - studying information systems development methodologies in practice. In

HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences, page 264.2. IEEE Computer Society, 2005.

[51] Marc Hassenzahl, Michael Burmester, and Franz Koller. AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In G. Szwillus and J. Ziegler, editors, *Mensch and Computer 2003: Interaktion in Bewegung*, pages 187–196, Stuttgart, 2003. B. G. Teubner.

[52] Orit Hazzan and James E. Tomayko. Human aspects of software engineering: The case of extreme programming. In *XP*, pages 303–311, 2004.

[53] Rashina Hoda, James Noble, and Stuart Marshall. Negotiating contracts for agile projects: A practical perspective. In *XP*, volume 31 of *Lecture Notes in Business Information Processing*, pages 186–191. Springer, 2009.

[54] P. Hodgetts. Experiences integrating sophisticated user experience design practices into agile processes. In *Agile Conference, 2005*, pages 235–242, 2005.

[55] Tasha Hollingsed and David 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*, pages 249–255, New York, NY, USA, 2007. ACM.

[56] Andreas Holzinger. Rapid prototyping for a virtual medical campus interface. *IEEE Softw.*, 21(1):92–99, 2004.

[57] Andreas Holzinger. Usability Engineering for Software Developers. *Communications of the ACM*, 48(1):71–74, January 2005.

[58] Andreas Holzinger, Maximilian Errath, Gig Searle, Bettina Thurnher, and Wolfgang 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*, pages 169–172, Washington, DC, USA, 2005. IEEE Computer Society.

[59] Andreas Holzinger and Wolfgang Slany. (XP+UE→XU) praktische erfahrungen mit extreme usability. *Informatik Spektrum*, 29(2):91–97, 2006.

[60] Jason I. Hong and James 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*, pages 717–724, New York, NY, USA, 2001. ACM Press.

[61] William Hudson. A tale of two tutorials: a cognitive approach to interactive system design and interaction design meets agility. *interactions*, 12(1):49–51, 2005.

[62] Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Martin Umgeher, and Thomas Vlk. Optimizing Extreme Programming. In *ICCCE 2008: Proceedings of the International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia*, pages 1052–1056. IEEE, 2008.

[63] Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Martin Umgeher, Thomas Vlk, Christina Koeffel, Manfred Tscheligi, and Peter Wolkerstorfer. Practical usability in XP software development processes. In *Journal of Universal Computer Science*, 2010. Submitted.

[64] Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Martin Umgeher, Thomas Vlk, and Peter Wolkerstorfer. User interface design for a content-aware mobile multimedia application: An iterative approach. In *MOMIR 2007 Frontiers in Mobile and Web Computing: Proceedings of MoMM2007 & iiWAS2007 Workshops*, volume 231, pages 115–120, Jakarta, Indonesia, December 2007. Second International Workshop on Mobile Multimedia Information Retrieval (MoMIR), 3-5 December, 2007, Jakarta, Indonesia ISBN : 978-3-85403-231-1.

[65] 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, Sainte Luce, Martinique, France*, pages 189–194. IEEE Computer Society, February 10-15, 2008.

[66] Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Martin Umgeher, and Peter Wolkerstorfer. Concept and design of a contextual mobile multimedia content usability study. In *ACHI*

2009, Second International Conference on Advances in Computer-Human Interaction, February 1-7, 2009 - Cancun, Mexico. IEEE Computer Society. To appear.

[67] Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Martin Umgeher, and Peter Wolkerstorfer. Integrating Extreme Programming and User-Centered Design. In *PPIG 2008, The 20th Annual Psychology of Programming Interest Group Conference, Lancaster University, UK. 10th - 12th September 2008*.

[68] Zahid Hussain, Martin Lechner, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Martin Umgeher, and Peter Wolkerstorfer. Agile User-Centered Design Applied to a Mobile Multimedia Streaming Application. In *USAB 2008*, volume 5298/2008 of *LNCS*, pages 313–330. Springer Berlin / Heidelberg, November 2008.

[69] Zahid Hussain, Harald Milchrahm, Sara Shahzad, Wolfgang Slany, Manfred Tscheligi, and Peter Wolkerstorfer. Integration of extreme programming and user-centered design: Lessons learned. In Pekka Abrahamsson, Michele Marchesi, and Frank Maurer, editors, *XP 2009*, volume 31 of *LNBIP*, pages 174–179. Springer, 2009.

[70] Zahid Hussain and Wolfgang 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*, pages 401–404. ACM, 2010.

[71] Zahid Hussain, Wolfgang Slany, and Andreas Holzinger. Current state of agile user-centered design: A survey. In *USAB 2009*, volume 5889 of *Lecture Notes in Computer Science*, pages 416–427. Springer Berlin / Heidelberg, November 2009. USAB 2009 - HCI and Usability for e-Inclusion, Linz, Austria, 9-10 November 2009.

[72] Zahid Hussain, Wolfgang Slany, and Andreas Holzinger. Investigating agile user-centered design in practice: A grounded theory perspective. In *USAB 2009*, volume 5889 of *Lecture Notes in Computer Science*, pages 279–289. Springer Berlin / Heidelberg, November 2009. USAB 2009 - HCI and Usability for e-Inclusion, Linz, Austria, 9-10 November 2009.

[73] Zahid Hussain, Wolfgang Slany, and Andreas Holzinger. Agile software development methods and usability/user-centered design: Perspectives

from a survey. In *Empirical Software Engineering Journal*. Springer, 2010. submitted.

[74] IpsosInsight. Mobile phones could soon rival the PC as world's dominant internet platform, April 2006. http://www.ipsos-na.com/news/pressrelease.cfm?id=3049, Last visited 24th April 2009.

[75] ISO 13407. Human Centred Design Process for Interactive Systems. International Organization for Standardization, Geneva, 1999.

[76] ISO 9241-11. Ergonomic requirements for office work with visual display terminals - part 11. International Organization for Standardization, Geneva, 1998.

[77] Melody 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*, pages 329–330, New York, NY, USA, 2000. ACM.

[78] Melody Y. Ivory and Marti A. Hearst. The state of the art in automated usability evaluation of user interfaces. Technical Report UCB/CSD-00-1105, EECS Department, University of California, Berkeley, Jun 2000.

[79] Melody Y. Ivory and Marti A. Hearst. Improving web site design. *IEEE Internet Computing*, 6(2):56–63, 2002.

[80] Myoung Hoon Jeon, Dae Yol Na, Jung Hee Ahn, and Ji Young Hong. User segmentation & ui optimization through mobile phone log analysis. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 495–496, New York, NY, USA, 2008. ACM.

[81] Bill Jerome and Rick Kazman. Surveying the solitudes: An investigation into the relationships between human computer interaction and software engineering in practice. In *Human-Centered Software Engineering Integrating Usability in the Software Development Lifecycle*. Springer Netherlands, 2005.

[82] Yong Gu Ji and Myung Hwan Yun. Enhancing the minority discipline in the IT industry: A survey of usability and User-Centered design practice. *International Journal of Human-Computer Interaction*, 20(2):117 – 134, 2006.

[83] Timo Jokela and Pekka Abrahamsson. Usability assessment of an extreme programming project: Close co-operation with the customer does not equal to good usability. In *5th International Conference, PROFES '04*, pages 393–407, 2004.

[84] D. Kane. Finding a place for discount usability engineering in agile development: throwing down the gauntlet. In *Agile Development Conference, 2003. ADC 2003. Proceedings of the*, pages 40–46, 2003.

[85] Eeva Kangas and Timo Kinnunen. Applying user-centered design to mobile application development. *Commun. ACM*, 48(7):55–59, 2005.

[86] Bill Krebs. http://agile.csc.ncsu.edu/survey/shodan_survey.html. Visited on 04.01.2008.

[87] Amy Law and Raylene Charron. Effects of agile practices on social factors. In *HSSE '05: Proceedings of the 2005 workshop on human and social factors of software engineering*, volume 30, pages 1–5, New York, NY, USA, July 2005. ACM Press.

[88] Jason Chong Lee. Embracing agile development of usable software systems. In *CHI '06 extended abstracts on Human factors in computing systems*, pages 1767–1770. ACM, 2006.

[89] J.C. Lee, D. Scott McCrickard, and K.T. Stevens. Examining the foundations of agile usability with extreme scenario-based design. In *Agile Conference, 2009. AGILE '09.*, pages 3 –10, 2009.

[90] Michael Leitner, Peter Wolkerstorfer, Reinhard Sefelin, and Manfred Tscheligi. Mobile multimedia: identifying user values using the means-end theory. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 167–175, Amsterdam, The Netherlands, 2008. ACM.

[91] Manifesto. Manifesto for agile software development. http://agilemanifesto.org/, 2001.

[92] Joe Marasco. Software development productivity and project success rates: Are we attacking the right problem? http://www.ibm.com/developerworks/rational/library/feb06/marasco/index.html, Feb 2006. last visited: 15-01-2008.

[93] Paul McInerney and Frank Maurer. UCD in agile projects: dream team or odd couple? *Interactions*, 12(6):19–23, 2005.

[94] Marc McNeill. User centred design in agile application development. http://www.thoughtworks.com/pdfs/agile_and_UCD_MM.pdf.

[95] Thomas Memmel, Harald Reiterer, and Andreas Holzinger. Agile methods and visual specification in software development: A chance to ensure universal access. In Constantine Stephanidis, editor, *HCII Universal Access in HCI*, volume 4554 of *LNCS*, pages 453–462. Springer, 2007.

[96] John Mendonca and Jeff Brewer. Lean, light, adaptive, agile and appropriate software development: the case for a less methodical methodology. pages 42–52, 2003.

[97] G. Meszaros and J. Aston. Adding usability testing to an agile project. In *Agile Conference, 2006*, 2006.

[98] L. Miller. Case study of customer input for a successful product. In *Agile Conference, 2005*, pages 225–234, 2005.

[99] Lynn Miller and Desirée Sy. Agile user experience SIG. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems CHI*, pages 2751–2754, Boston, MA, USA, 2009. ACM.

[100] Subhas Chandra Misra, Vinod Kumar, and Uma Kumar. Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11):1869–1890, 2009.

[101] Rolf Molich and Jakob Nielsen. Improving a human-computer dialogue. *Commun. ACM*, 33(3):338–348, 1990.

[102] Elden Nelson. Extreme programming vs. interaction design, 2002. FTP Online.

[103] Jacob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993.

[104] James E. Nieters, Subbarao Ivaturi, and Iftikhar Ahmed. Making personas memorable. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 1817–1824, New York, NY, USA, 2007. ACM.

[105] Nielsen Norman. Agile usability: Best practices for user experience on agile development projects. Nielsen Norman Group Report, 2009. http://www.nngroup.com/reports/agile/.

[106] Hartmut Obendorf and Matthias Finck. Scenario-based usability engineering techniques in agile development processes. In *CHI '08*, pages 2159–2166. ACM, 2008.

[107] 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*, pages 439–445, New York, NY, USA, 2002. ACM.

[108] Fabio Patern, Angela Piruzza, and Carmen Santoro. Remote web usability evaluation exploiting multimodal information on user behavior. In Galle Calvary, Costin Pribeanu, Giuseppe Santucci, and Jean Vanderdonckt, editors, *CADUI*, pages 287–298. Springer, 2006.

[109] Jeff Patton. Hitting the target: adding interaction design to agile software development. In *OOPSLA 2002 Practitioners Reports*, Seattle, Washington, 2002. ACM.

[110] Jeff Patton. Twelve emerging best practices for adding UX work to agile development, June 2008. http://www.agileproductdesign.com/blog/emerging_best_agile_ux_practice.html.

[111] Eric Paulos, August Joki, Parul Vora, and Anthony Burke. Anyphone: mobile applications for everyone. In *DUX '08: Proceedings of the 2007 conference on Designing for User eXperiences*, pages 2–8, New York, NY, USA, 2007. ACM.

[112] Marc Pifarré and Oscar Tomico. Bipolar laddering (bla): a participatory subjective exploration method on user experience. In *DUX '07: Proceedings of the 2007 conference on Designbiolaring for User eXperiences*, pages 2–13, New York, NY, USA, 2007. ACM.

[113] John Pruitt and Tamara Adlin. *The Persona Lifecycle: Keeping People in Mind Throughout Product Design (The Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[114] Vinay Ramachandran and Anuja Shukla. Circle of life, spiral of death: Are xp teams following the essential practices? In *Extreme Programming and Agile Methods XP/Agile Universe 2002*. Springer, 2002.

[115] Thyra Rauch and Tom Wilson. UPA and CHI surveys on usability processes. *SIGCHI Bulletin*, 27(3), 1995.

[116] Donald J. Reifer. How good are agile methods? *IEEE Software*, 19(4):16–18, 2002.

[117] Thomas J. Reynolds and Jerry C. Olson, editors. *Understanding Consumer Decision Making: The Means-end Approach to Marketing and Advertising Strategy*. Lawrence Erlbaum, Mahwah, NJ, 2001.

[118] Stephanie Rosenbaum, Janice Anne Rohn, and Judee Humburg. A toolkit for strategic usability: results from workshops, panels, and surveys. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2000.

[119] Virpi Roto. *Web Browsing on Mobile Phones – Characteristics of User Experience*. PhD thesis, Helsinki University of Technology, Finland, 2006.

[120] Virpi Roto and Eija Kaasinen. The second international workshop on mobile internet user experience. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 571–573, New York, NY, USA, 2008. ACM.

[121] Jeffrey Rubin. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[122] Bernhard Rumpe and Astrid Schröder. Quantitative survey on extreme programming projects. In *Extreme Programming and Agile Methods XP/Agile Universe 2002*, 2002.

[123] O. Salo and P. Abrahamsson. Agile methods in european embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum. *Software, IET*, 2, 2008.

[124] R.S. Schreiber. The 'how to' of grounded theory: avoiding the pitfalls. In Noerager Stern P. Schreiber, R.S., editor, *Using Grounded Theory in Nursing*. Springer, Berlin, 2001.

[125] Carolyn B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Trans. Softw. Eng.*, 25(4):557–572, 1999.

[126] Ahmed Seffah, Jan Gulliksen, and Michel C. Desmarais. *Human-Centered Software Engineering - Integrating Usability in the Development Process (Human-Computer Interaction Series)*. Springer-Verlag New York, Inc., 2005.

[127] Sara Shahzad, Zahid Hussain, Martin Lechner, and Wolfgang Slany. Inside view of an extreme process. In *Agile Processes in Software Engineering and Extreme Programming*, LNBIP, pages 226–227. Springer Verlag, 06 2008. 9th International Conference, XP 2008, Limerick, Ireland, June 10-14, 2008.

[128] Lori Shyba and James 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*, pages 187–194, New York, NY, USA, 2005. ACM.

[129] A. Sillitti, M. Ceschi, B. Russo, and G. Succi. Managing uncertainty in requirements: a survey in documentation-driven and agile companies. In *Software Metrics, 2005. 11th IEEE International Symposium*, 2005.

[130] Anselm Strauss and Juliet M. Corbin. *Basics of Qualitative Research : Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, 1st edition, 1990.

[131] Anselm Strauss and Juliet M. Corbin. *Basics of Qualitative Research : Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, 2nd edition, 1998.

[132] S.R. Subramanya and Byung K. Yi. User interfaces for mobile content. *IEEE Computer*, 39(4):85–87, April 2006.

[133] D. Sy. Adapting usability investigations for agile user-centered design. *Journal of Usability Studies*, 2(3):112132, 2007.

[134] Desiree Sy. Coherent agile user-centered design. UPA 2009 International Conference. Usability Professional's Association, 2009.

[135] Bjørnar Tessem. Experiences in learning xp practices: A qualitative study, xp2003. In *XP*, pages 131–137, 2003.

[136] J. Ungar. The design studio: Interface design for agile teams. In *Agile, 2008. AGILE '08. Conference*, pages 519–524, 2008.

[137] Usability.gov. Step-by-Step Usability Guide. http://www.usability.gov/. Last visited: 18.08.2008.

[138] Heli Väätäjä. Factors affecting user experience in mobile systems and services. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 551–551, New York, NY, USA, 2008. ACM.

[139] Karel Vredenburg, Ji-Ye Mao, Paul W. Smith, and Tom Carey. A survey of user-centered design practice. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2002.

[140] Ljiljana Vukelja, Lothar Müller, and Klaus Opwis. Are engineers condemned to design? a survey on software engineering and ui design in Switzerland. In *INTERACT (2)*, volume 4663 of *LNCS*, pages 555–568. Springer, 2007.

[141] W3C. Notes on user centred design process (UCD). http://www.w3.org/WAI/EO/2003/ucd, April 2004. Last visited: 19.01.2009.

[142] Sarah J. Waterson, Jason I. Hong, Tim Sohn, James A. Landay, Jeffrey Heer, and Tara Matthews. What Did They Do? Understanding Clickstreams with the WebQuilt Visualization System. In *AVI '02: Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 94–102, New York, NY, USA, 2002. ACM.

[143] Cathleen Wharton, John Rieman, Clayton Lewis, and Peter Polson. The cognitive walkthrough method: a practitioner's guide. pages 105–140, 1994.

[144] Elizabeth Whitworth and Robert Biddle. The social nature of agile teams. In *AGILE '07: Proceedings of the AGILE 2007*, pages 26–36. IEEE Computer Society, 2007.

[145] Peter Wolkerstorfer, Manfred Tscheligi, Reinhard Sefelin, Harald Milchrahm, Zahid Hussain, Martin Lechner, and Sara Shahzad. Probing an agile usability process. In *CHI '08: human factors in computing systems*, pages 2151–2158, New York, USA, 2008. ACM.

[146] Ronggang Zhou, Shengshan Huang, Xiangang Qin, and J. Huang. A survey of user-centered design practice in China. In *IEEE International Conference on Systems, Man and Cybernetics, SMC 2008*, pages 1885–1889, 2008.

Appendix

Appendix: Agile UCD Survey Questionnarie

Agile User-Centered Design Questionnaire 2009

The goal of this questionnaire is to analyze the integration of agile development methods – particularly Extreme Programming (XP) and User-Centered Design (UCD) methodologies. Our hope is to gain a better understanding of what is actually happening on real-world agile projects and the benefits and limitations of the integrated process.

Organization of the Questionnaire:

Page 1 – Questionnaire Overview (current page)
Page 2 – About you and the company / Project (8 questions)
Page 3 – Agile User-Centered Design (20 questions)

It should take you about 5-10 minutes to complete this questionnaire. Thank you very much for taking your valuable time to fill it out.

Statement of Confidentiality and Disclosure:
All responses will be treated anonymously and all data will be held strictly confidential. The data obtained from this questionnaire will be used solely for the purpose of research and will be used in a doctoral dissertation. The anonymous results may possibly appear in an academic publication.

There are 28 questions in this survey.

Next >> [Exit and clear survey]

Agile User-Centered Design Questionnaire 2009

0% [ ] 100%

About you and the company/Project

Q1. What job function best describes your role?

Choose one of the following answers

- ○ Executive / Director
- ○ Project / Program Manager
- ○ Developer / Software Engineer / Programmer
- ○ Usability Engineer / UI/UX/Interaction Designer
- ○ Consultant
- ○ Systems Administrator
- ○ Database Administrator
- ○ QA / Testing
- ○ Other [ ]
- ◉ No answer

Q2. How many years of experience do you have working in software development?

Choose one of the following answers

- ○ 1 Year
- ○ 2 – 5 Years
- ○ 6 – 10 Years
- ○ 11 – 20 Years
- ○ > 20 Years
- ◉ No answer

Q3. How many years have you been working in agile software development methods?

Choose one of the following answers

- ○ 1 Year
- ○ 2 – 5 Years
- ○ 6 – 10 Years
- ○ 11 – 20 Years
- ○ > 20 Years
- ◉ No answer

Q4. Which Agile methods are you working in?

Check any that apply

- ☐ Extreme Programming (XP)
- ☐ SCRUM
- ☐ DSDM
- ☐ Lean Development
- ☐ Crystal Methods
- ☐ Adaptive Software Development
- ☐ Pragmatic Programming
- ☐ Agile Unified Process/ Open UP
- ☐ Other: [ ]

Q5. Which industry sector best describes your organization?

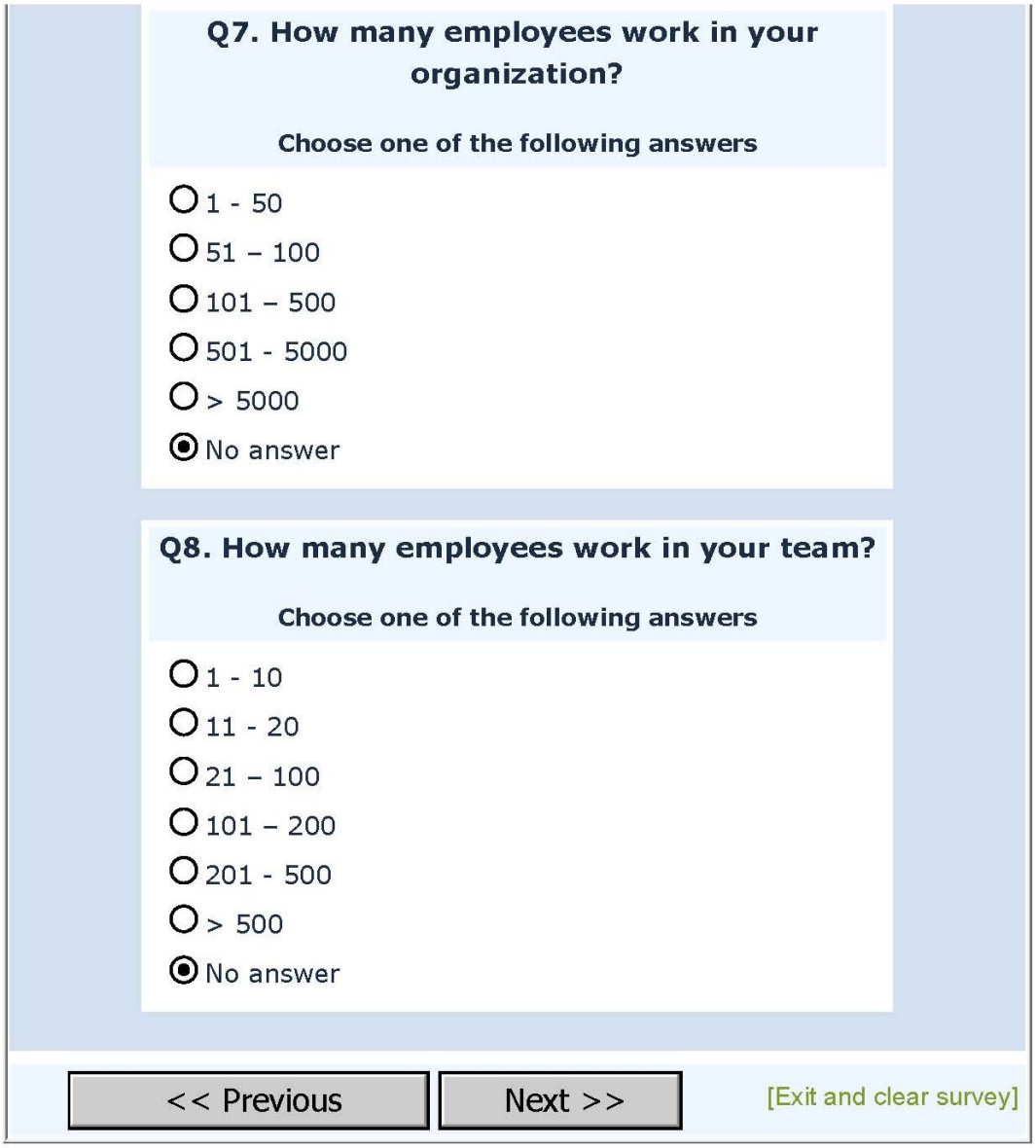
Choose one of the following answers

- ○ Financial
- ○ Construction
- ○ Education
- ○ Government
- ○ Manufacturing
- ○ Health
- ○ Software
- ○ Telecommunications
- ○ IT Services
- ○ Transportation
- ○ Not for Profit Organization
- ○ Agricultural
- ○ Distribution
- ○ Real Estate
- ○ Retail
- ○ Other
- ◉ No answer

Q6. Where are you based?

Choose one of the following answers

- ○ North America
- ○ Europe
- ○ Asia
- ○ South & Central America
- ○ Australia & New Zealand
- ○ Africa
- ◉ No answer



Q7. How many employees work in your organization?

Choose one of the following answers

- ○ 1 - 50
- ○ 51 – 100
- ○ 101 – 500
- ○ 501 - 5000
- ○ > 5000
- ◉ No answer

Q8. How many employees work in your team?

Choose one of the following answers

- ○ 1 - 10
- ○ 11 - 20
- ○ 21 – 100
- ○ 101 – 200
- ○ 201 - 500
- ○ > 500
- ◉ No answer

<< Previous | Next >> | [Exit and clear survey]

Agile User-Centered Design Questionnaire 2009

0% 100%

About Agile User-Centered Design Methodologies

Q9. Please choose some of the agile practices adopted by your team and the roles carried out in your team.

Check any that apply

- ☐ High Level Release Planning
- ☐ Iteration / Sprint Planning
- ☐ Daily Stand-up / Scrum Meeting
- ☐ Sit Together
- ☐ Pair Programming
- ☐ Test Driven Development
- ☐ Refactoring
- ☐ Continuous Integration
- ☐ Automated Acceptance Testing
- ☐ Collective Code Ownership
- ☐ Coding Standards
- ☐ Sustainable Pace
- ☐ Simple Design
- ☐ Stories
- ☐ Incremental Design
- ☐ Small Releases
- ☐ Retrospectives
- ☐ Status Reports
- ☐ On Site Customer
- ☐ Customer is an End-User
- ☐ Active Business Analyst / Product Owner / Customer Participant
- ☐ Agile Coach Present in Team
- ☐ Other:

Q10. Does your team use a regular iteration or sprint schedule? If so, how often are your iterations?

Choose one of the following answers

- ○ No iterations
- ○ Weekly
- ○ Every two weeks
- ○ Every three weeks
- ○ Monthly
- ○ Every six weeks
- ○ Other
- ◉ No answer

Q11. If your team uses iterations or sprints, are they fixed in length (also called time-boxed) or do they vary?

Choose one of the following answers

- ○ Fixed
- ○ Variable
- ○ Other
- ◉ No answer

Q12. How is the role of a Usability Engineer / UI/ Interaction Designer carried out in your organization?

Choose one of the following answers

- ○ Usability Engineer / UI/Interaction Designer co-located with team
- ○ Usability Engineer / UI/Interaction Designer not co-located with team but present at remote site
- ○ No Usability Engineer / UI/Interaction Designer present in the team but this role is performed by a developer having interest in HCI / Usability
- ○ No role for a Usability Engineer / UI/Interaction Designer present in the team at all
- ○ Other

◉ No answer

Q13. If a Usability Engineer / UI/Interaction Designer is present in your team, in your opinion which approach is better: co-located vs. remotely located Usability Engineer / UI/Interaction Designer?

Choose one of the following answers

Please choose one of the following:

Please enter your comment here:

○ Approach of co-located UE / UI/Interaction Designer is better

○ Approach of remotely located UE / UI/Interaction Designer is better

◉ No answer

Q14. We find the following communication channels for conveying information with the Usability Engineer / UI/Interaction Designer to be:

	Very Effective	Effective	Neutral	Ineffective	Very Ineffective	Don't Know	No answer
Face-to-Face (F2F) Communication	○	○	○	○	○	○	◉
Email	○	○	○	○	○	○	◉
Online Chat	○	○	○	○	○	○	◉
Teleconference Calls	○	○	○	○	○	○	◉
Video Conferencing	○	○	○	○	○	○	◉

Q15. If in your team the role of Usability Engineer / UI/ Interaction Designer is performed by a developer having interest in HCI / usability; then in your opinion which approach is better: Dedicated Usability Engineer / UI/Interaction Designer present in the team or this role performed by a developer?

Choose one of the following answers

Please choose one of the following:

- ○ Dedicated UE / UI/ Interaction Designer present in the team is better
- ○ Role of UE / UI/ Interaction Designer performed by a developer is better
- ◉ No answer

Please enter your comment here:

Q16. Do you think that it's possible that a developer can pick-up HCI skills?

Choose one of the following answers

- ○ Definitely
- ○ Nearly
- ○ Sometimes
- ○ Rarely
- ○ Never
- ◉ No answer

Q17. How can a developer learn HCI skills?

Check any that apply

- ☐ By Pairing with a Usability Engineer / UI/Interaction Designer
- ☐ By Training
- ☐ Other: [ ]

Q18. What types of HCI instruments are used in your team/project?

Check any that apply

- ☐ Field Studies
- ☐ Observational Studies of Users
- ☐ Conceptual Designs
- ☐ Personas
- ☐ Low-Fidelity Prototyping
- ☐ Laboratory Usability Testing
- ☐ Remote Usability Testing
- ☐ Usability Expert Evaluations
- ☐ Conceptual Inquiry
- ☐ Ethnographic Research
- ☐ Rapid Iterative Testing
- ☐ Needs Analysis
- ☐ Goal-Directed Design
- ☐ Automated Usability Evaluations
- ☐ Other: [ ]

Q19. When do HCI / usability evaluations occur in your project?

Check any that apply

- ☐ Before the project begins
- ☐ During the project (all phases)
- ☐ During the project (acceptance test phase)
- ☐ Missing
- ☐ Don't know

Q20. Do the requirement details emerge over time on your projects or are they fixed up-front?

Choose one of the following answers

- ○ Emerge over time
- ○ Fixed up-front
- ◉ No answer

Q21. If the requirement details emerge over time then do you do some UI design up-front before any development begins in your projects?

Choose one of the following answers

Please choose one of the following:

- ○ Yes
- ○ No
- ◉ No answer

Please enter your comment here:

Q22. Over the past year how often has your team released new versions of the software to end-users?

Q23. Over the next year how often does your team expects to release new versions of the software to end-users?

Q24. Once something is released, it is common in agile teams to release modifications, updates, or extensions based on feedback or usage data.
For your projects, what is the usual time between releasing something and then releasing feedback-driven change to it?

Choose one of the following answers

- ○ Week
- ○ 2 Weeks
- ○ Month
- ○ 3 Months
- ○ Other
- ◉ No answer

Q25. The adoption of the agile user-centered process by our team has resulted in the improvement of usability and quality of the software that we develop.

Choose one of the following answers

Please choose one of the following:

- ○ Strongly Agree
- ○ Agree
- ○ Neutral
- ○ Disagree
- ○ Strongly Disagree
- ○ Don't Know
- ◉ No answer

Please enter your comment here:

Q26. Due to the agile user-centered design process adopted by our team, the resulting software product has increased the satisfaction of its end-users.

Choose one of the following answers

Please choose one of the following:

- ○ Strongly Agree
- ○ Agree
- ○ Neutral
- ○ Disagree
- ○ Strongly Disagree
- ○ Don't Know
- ◉ No answer

Please enter your comment here:

Q27. The integration of agile method(s) with user-centered design / usability has added value to our adopted process and to our team.

Choose one of the following answers

Please choose one of the following:

- ○ Strongly Agree
- ○ Agree
- ○ Neutral
- ○ Disagree
- ○ Strongly Disagree
- ○ Don't Know
- ◉ No answer

Please enter your comment here:

Q28. Any aspect that has not been covered by this questionnaire or any comments you would like to add, please describe here shortly.

<< Previous | Submit | [Exit and clear survey]

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …… ……..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…… ……..
date (signature)