Open Research Online

The Open University's repository of research publications
and other research outputs

User experience design and agile development : integration as an on-going achievement in practice

Thesis

For guidance on citations see FAQs.

Version: Version of Record

oro.open.ac.uk

The Open University

User Experience Design and Agile Development:
Integration as an on-going achievement in practice

A thesis submitted to The Open University

for the degree of

Doctor of Philosophy in Computing

by

Jennifer Ferreira

2011

DATE OF SUBMISSION : 15 AUGUST 2011

DATE OF AWARD : 8 AUGUST 2012

Abstract

This research investigates how Agile development is combined with User Experience (UX) design. Agile development and UX design have roots in different disciplines and practitioners have to reconcile their perspectives on developing software if they are to work together. To date, there has been no sustained academic study on how Agile developers and UX designers work together in practical settings on a day-to-day basis. The ethnographically-informed research in this dissertation consists of three studies of teams in organisational settings, combined with an analysis of accounts of Agile development and UX design practice found in the literature. Together, they provide evidence for the complex, multifaceted nature of the work that integrates Agile development with UX design.

The studies of day-to-day practice conducted for this research, found the work of the Agile developers and UX designers to be localised, contingent and purposeful. Agile development and UX design integration, as it was achieved in the teams studied, was negotiated and achieved on a day-to-day basis between the developers and designers. The findings from the analysis of accounts of practice from the literature show that integration is achieved with the right tools, techniques and processes that coordinate between the tasks of the developers and designers and establish a focus on usability and on releasing working software. However, the accounts contain little and conflicting evidence for what constitutes the day-to-day work of Agile developers and UX designers in practical settings and as a result the utility of tools, techniques and processes for practice is not clear.

Informed by the findings from the accounts in the literature and the studies of practice, five facets emerged as integral to an understanding of how the integration of Agile development and UX design is an on-going achievement in practice. These facets are (1) focus and coordination, (2) mutual awareness, (3) expectations about acceptable behaviour, (4) negotiating progress and (5) engaging with each other. The extent to which these facets enable integration, depend on contextual values concern-

1

ing the combination of Agile development and UX design endorsed in the organisation. These findings serve to establish conditions which can constrain and enable Agile developers and UX designers in their integration work, while being sympathetic to the values embedded in the settings in which they work.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

There is on-going debate among academics and practitioners about how Agile development relates to User Experience (UX) design, and how they could be combined. UX design and Agile development have roots in different disciplines, bringing with them different perspectives on software development. This presents challenges for developers and designers working together. While the growing body of literature continues to focus on process descriptions and recommended techniques, the day-to-day work involved and the many and varied settings in which the techniques and processes are applied, remain largely unexplored.

This dissertation asks "How are UX design and Agile development combined?" To address this question a qualitative, ethnographically-informed approach was adopted.

The combination of Agile development with UX design was investigated both in terms of how the Agile/UX community writes about its experiences and what happens in practice. Fifty two accounts of combining Agile development and UX design practice were strategically selected from the literature and analysed thematically. Three field studies were conducted with small to medium-sized Agile teams based in the UK. Data collected via observations and interviews was analysed thematically.

As a result of independent analyses, the findings from accounts of practice and findings from the field work provide evidence for the complex, multifaceted nature of the work that integrates Agile development with UX design. The results from this research show that combining Agile development with UX design in practice is a highly localised, contingent solution that brings improvements for specific circumstances. Every setting brings unique challenges and shapes the work practitioners do, and hence shapes the integration of Agile development with UX design. The understanding gained from the field studies and the analysis of the accounts of practice in the literature, suggests that improving practice requires further explication of contextual issues that shape practice, such as values and assumptions underlying decision-making and work culture.

1.1 Background and terminology

The next sections provide background on the features of Agile software development (§ 1.1.1), User Experience (UX) design (§ 1.1.2) and other terminology considerations that are relevant for the discussion in this dissertation.

1.1.1 Agile software development

This section introduces features of *Agile software development* that are relevant for the discussion in this dissertation, i.e., Agile software development as a collection of software development approaches that adhere to a coherent set of values and principles, that are people-driven, and deliver working software via iterative and incremental

development.

Agile values

In 2001 a group of leading software development practitioners agreed on the Agile Manifesto — a coherent set of values and principles which underpin approaches to software development now commonly referred to as *Agile* methods. The value statement is as follows:

> "We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> - **Individuals and interactions** over processes and tools
> - **Working software** over comprehensive documentation
> - **Customer collaboration** over contract negotiation
> - **Responding to change** over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more."[1]

Signatories included representatives of Agile methods such as eXtreme Programming (XP) [Beck 1999; Beck and Andres 2004], Scrum [Schwaber and Beedle 2002], Crystal Clear [Cockburn 2004] and Feature Driven Development (FDD) [Palmer and Felsing 2002]. As the Agile Manifesto articulates, the emphasis of these methods is on people, working software and responding to change. Agile methods are designed to deal with the "change, speed and uncertainty" [Sharp and Robinson 2004] that trouble projects following *plan-driven approaches*. Fowler [2005] has characterised plan-driven approaches as those that "impose a disciplined process upon software development with the aim of making software development more predictable and more efficient." Contrasting with the plan-driven approach is the flexibility of Agile development's *adaptive approach*:

---

[1] http://agilemanifesto.org/.

3

Agility, for a software development organisation, is the ability to adapt and react expeditiously and appropriately to changes in its environment and to demands imposed by this environment. An Agile process is one that readily embraces and supports this degree of adaptability. So, it is not simply about the size of the process, or the speed of delivery; it is mainly about flexibility. [Kruchten 2001].

Agile development embraces change rather than restraining changes from the customer with early commitments and adhering to plans drawn up at the outset of the development project [Beck 1999; Beck and Andres 2004].

People-driven

Agile development is people-driven in the sense that people and the interactions between them are recognised as vital to project success. Lindstrom and Jeffries [2004] explain that "The determining factor of project success seemed more and more to be the people on the project, not the technology or the methods that were being used." A review of project methodologies in practice revealed that "People's characteristics are a first-order success driver, not a second-order one." [Cockburn 1999]. Cockburn and Highsmith [2001] list people factors that Agile development relies on as " amicability, talent, skill, and communication." Supporting these factors is a major concern for the Agile team.

Agile teams achieve their work through self-organisation and collaboration [Cockburn and Highsmith 2001]. That is, in Agile development the developers are considered to be responsible professionals who are in the best position to take decisions regarding how to accomplish their work. For example, XP requires that the developers are responsible for technical decisions, as well as the estimates for how long the work will take [Fowler 2005]. Lindvall et al. [2002] point out that teams require some flexibility in how the work is done, including the appropriate organisational support "In addition, teams need some amount of local control; they must have the ability

4

to adapt working practices as they feel appropriate." Agile practices support intense collaboration within the Agile team and with the customer [Highsmith and Cockburn 2001]. Relying on the team's tacit knowledge [Boehm and Turner 2005] places less emphasis on documentation and other practices that add unnecessary weight to the process, or slows development down.

Working software via iterative and incremental development

Agile development proceeds iteratively and incrementally. Iterative development provides Agile teams with regular intervals for obtaining feedback about the process and the product. Based on this feedback, the team can determine whether the process and the product are still applicable to the current situation and adjust as necessary [Fowler 2005]. An Agile development iteration is illustrated in figure 1.1. At the beginning of the iteration a set of requirements, or features, is selected and prioritised with the customer[2], after which the developers set about implementing those features. The implementation work only lasts for the length of one iteration — from one week to one month — at the end of which the implemented product is evaluated, along with the process. The team assess the accuracy of the work estimates created at the iteration planning and ask questions such as: "What did the team do well?"; "What can be improved?"; etc. [Derby et al. 2006]. If, at the end of the iteration, there are still outstanding features to be implemented, then the next iteration is planned and carried out and repeated until the customer agrees that the required features have been implemented. "In incremental development, the repetitions of activities address new parts of the system" [Cockburn 1993] and so Agile development employs incremental development by successively adding the next highest priority features. At the end of each iteration, Agile teams are expected to have tested, working software that can be shown to a customer [Abrahamsson et al. 2002]. One of the Agile principles states that "Working software is the primary measure of progress."

---

[2]The customer may be a potential end user of the product under development or a business representative from the client company.
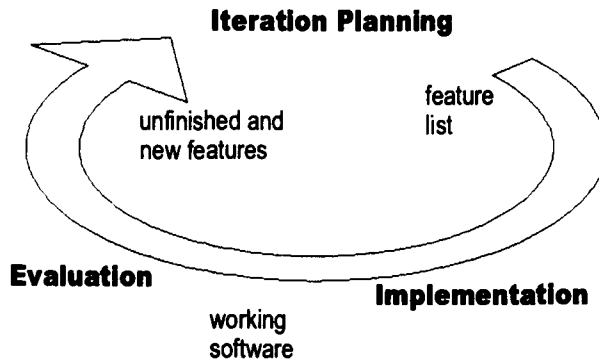
Figure 1.1: Agile iteration

Agile terminology for this dissertation

Use of the Agile terminology in this dissertation closely follows the way in which accounts found in the literature employ these terms:

Agile software development Or **Agile development**. In reviewing the literature "Agile development" is found to be used as an umbrella term for agile methodologies (e.g. [Blomkvist 2005]) and is still referred to as such by the thought leaders in agile development (e.g. Jeff Sutherland [Sutherland 2010]). It is common for practitioners to write about "Agile projects" or "agile development" when their particular method was eXtreme Programming [Meszaros and Aston 2006; Patton 2002a] or Scrum [Ungar and White 2008]. When the term "Agile software development" is used in this dissertation, it is used as the umbrella term. In this dissertation, reference is made to the specific Agile methods, such as XP and Scrum, as appropriate.

Agile development team Or **Agile team**. Some practitioner reports use the term "Agile team" when referring to a team of developers using Scrum (e.g. [Illmensee and Muff 2009]). In the same way, this dissertation uses "Agile team" even when the specific development approach is known.

Agile developer Or **developer**. Where these terms appear in this dissertation,

they are referring to a developer involved in an Agile development method such as XP or Scrum, and mainly involved in the coding activities of software development. In this dissertation 'developer' or 'Agile developer' is used and the specific method or approach is referenced when that is relevant to the discussion.

Iteration Or **Sprint** Or **Cycle**. All three terms appear in the literature on Agile development. "Iteration" is generally associated with XP, while "sprint" is associated with Scrum. "Cycle" is also used. Whether iteration, sprint or cycle is used in this dissertation depends on whether it appears in a published text being referenced, or what the teams under study used in their everyday work. For example, participants in Study 1 talked about 'sprints' while participants in Study 3 talked about 'cycles'.

1.1.2 User experience design

This section introduces what is meant with *User Experience (UX) design* as it is used in this dissertation, i.e., as a collection of approaches that design the users' experience with the software by setting out to understand users and how they will use the software, and iteratively refining the design.

Designing the user experience

The term 'user experience design' has emerged as a way to escape the narrow view that those who design the user experience focus only on usability evaluations. Light [2006] explains that attempts at understanding the user experience represent a shift in how interactions between people and technology are analysed. User experience encompasses ideas ranging from "traditional usability to beauty, hedonic, affective or experiential aspects of technology use," [Hassenzahl and Tractinsky 2006], as opposed to "evaluating performance" [Light 2006]. The term has been adopted by the HCI community, but remains vague [Forlizzi and Battarbee 2004; Gulliksen et al. 2008; Hassenzahl and Tractinsky 2006]. Various definitions are offered in the literature and

McCarthy and Wright [2004] provide an in-depth investigation into how technology participates in the "felt experience" as people use and live with technology.

For the purposes of the discussion in this dissertation User Experience Design is used to refer to the collection of methods, tools, techniques, etc. for involving and maintaining focus on the end user in software development. Section 2.5.1 continues the discussion on the complexities surrounding design-related terminology in the context of the combination with Agile development.

Based on their observations of interaction design practice Sharp et al. [2007] model an interaction design lifecycle, which is adapted in figure 1.2. In this model, a software development project begins with activities that 'Identify needs/establish requirements.' Interaction designers may obtain data about needs and requirements from surveys, interviews, or in-situ observation sessions [Sharp et al. 2007]. In order to make sense of the data gathered during the user research activities, the interaction designers may model the information about the users using personas (user archetypes) [Cooper 1999] or scenarios (narrative descriptions of user tasks within a context) [Carroll 1995].

Based on the user research interaction designers may proceed to 'Build an interactive version' of the interaction design, which, regardless of the fidelity, may be evaluated. Evaluation may or may not include end users. UX designers may evaluate the user's experience with the designs in several ways. One way may be to have the user walk through a design, known as a *walkthrough* or *user review*, where the design could be in the form of a prototype [Constantine and Lockwood 1999]. Another evaluation technique may be to conduct a laboratory evaluation session, in which interaction with the software is controlled and statistical data about the structure, semantics and procedures within the user interface can be collected [Constantine and Lockwood 1999].

Iterative refinement

As the model in figure 1.2 shows, the interaction design process can cycle through stages of, for example, 'Build an interactive version', 'Evaluate' and '(Re)Design', thus iteratively developing interaction designs informed by evaluation feedback. When the maximum number of cycles through the model that resources will allow has been reached, or when the design is of the appropriate quality, the process concludes with the final product.
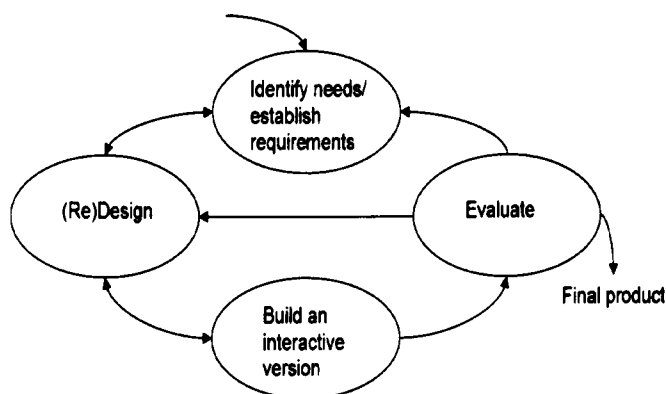
Figure 1.2: Interaction design lifecycle adapted from Sharp et al. [2007]

UX design terminology for this dissertation

The existence of many definitions and taxonomies in the Agile/UX literature, as well as practice, means it is not very useful to pin down one definition that captures all the design-related disciplines and approaches that are combined with Agile development. The following terms are working definitions to aid the rest of the discussion:

*design: In reviewing the literature, various design approaches that are combined with Agile development can be found. The following list contains a selection of these phrases:

> usage centered design [Patton 2002b]; user experience design [Hodgetts 2005]; user centered design [Miller 2005]; interaction design [Pat-

ton 2002a]; usability engineering [Sohaib and Khan 2010]; user interface design [Ferreira et al. 2007a].

"*design" is used as a collective term in this dissertation to refer to the various design approaches appearing in the literature (§ 2.5.1), without relabelling these approaches as something the authors did not intend, as for example, "UX design". Reference is made to the specific design approaches, such as User-Centered Design (UCD) or Interaction Design (ID), as appropriate.

UX design: Choosing "UX design" for this dissertation is in the spirit of encompassing the various *design disciplines and approaches that are combined with Agile development. Therefore, "UX design" is used as an umbrella term in the same way as "Agile development" is an umbrella term for the Agile methods.

UX designer Or **designer**. Where these terms appear in this dissertation, they are referring to a designer involved in UX design and mainly involved in carrying out the activities of UX design as defined in this section. In this dissertation, 'designer' or 'UX designer' is used and the specific method or approach is referenced when that is relevant to the discussion.

1.1.3 Combining Agile development and UX design

Reports in the literature present a variety of reasons *why* Agile development is combined with *design. For practitioners, the combination has advantages, e.g. maximising software quality, usability etc., it addresses some need in the organisation or the team's current software development effort, or the combination arises from an organisational or team transition to Agile development(e.g. [Budwig et al. 2009]). For academic researchers, the combination of Agile development and UX design is of interest as part of a broader, on-going effort for better integration of HCI into software engineering. Current research shows that it is far from smooth sailing for practitioners to achieve (see section 2.3) and there are challenges to be overcome in combining approaches with differing provenance.

Both Agile development and UX design aim to build quality software, but despite their common concern, each approaches development from a different perspective. While Agile methods mainly describe activities addressing code creation (e.g. [Beck and Andres 2004]), UX design methods describe activities for designing the product's interaction with a user [Sharp et al. 2007]. There is little guidance about integrating these two perspectives, and still few detailed accounts providing a close scrutiny of Agile development and UX design being combined in practice. Previous discussions of how User Experience (UX) designers and Agile developers can work together have focused on bringing the disciplines together by merging their processes or adopting specific techniques. To date, these discussions have either focused on integrating two separate processes (e.g. [Miller 2005; Patton 2003]), or on incorporating techniques from UX design into the Agile context (e.g. personas [Haikara 2007], or scenarios [Obendorf and Finck 2008]), or adapting techniques to fit an Agile development process (e.g. [Singh 2008]).

Combining Agile development and UX design terminology

User Experience Design and Agile development The first part of the title of this dissertation, "User Experience Design and Agile Development," has been chosen to represent the meeting of two disciplines that have and continue to evolve along separate trajectories. On the one hand, UX Design as an HCI-related discipline, and, on the other hand, Agile development with a firm grounding in software engineering. In this dissertation the distinction between Agile development and UX design is as follows: Agile development focuses on creating *working software*, while UX design focuses on creating a *usable design* that may or may not be in the form of working software.

Agile/UX literature The term 'Agile/UX literature' is short hand for the body of literature that contains accounts of the combination of Agile development and UX design.

1.2 Research questions

The overarching research question for this dissertation is:

How are UX design and Agile development combined?

The perspective taken in this dissertation, considers combining Agile development and UX design as a challenge that plays out in practice. In this perspective, understanding how they are (or can be) combined requires an understanding of how it is achieved by people engaging in Agile development and UX design practice. To address this question, selected accounts of practice from the literature were analysed and the results combined with studies of teams in organisational settings. The analysis of accounts of practice and the field studies generated more focused research questions. The questions addressed with the field work are:

What shapes the combination of Agile development and UX design in practice?

How is Agile development and UX design work accomplished?

Analysing selected accounts of practice in the literature explicates the various ways in which contributors to the literature conceptualise the combination of Agile development and UX design as a challenge and the solutions applied. The question that guided the analysis of the accounts of practice was:

What are the existing perspectives on combining Agile development and UX design in accounts of practice in the literature?

Which was further broken down into:

- How is the issue of combination conceptualised and how is it addressed?

- What is the experience of using Agile development and UX design together in practice?

12

1.3 Integration as an on-going achievement in practice: Overview of the findings

Next, an overview of the findings from the analysis of accounts of practice and the field work is given. The main themes from this research explain how the interplay of work setting and features of work give rise to a view on the integration of Agile development and UX design as an *on-going achievement in practice*.

Aspects of the practical settings that shape the work of the Agile developers and UX designers

In the analysis of accounts of practice, practitioner reports were found to present their settings as multidimensional, i.e., multiple teams and disciplines have to work together on software development projects. The accounts are not consistent in the details they provide about their settings, however, they tend to describe the product the project was aiming to develop and how the Agile developers were organised in relation to the designers. Empirical studies, a subset of the literature included in the analysis, were found to report few findings in terms of how the settings in which Agile development and UX design are combined, shape how that combination is achieved.

The field work conducted for this dissertation, found that the wider organisational setting in which the developers and designers are embedded has consequences for how they get their work done. This was observed in terms of the separation that was maintained between the developers and designers in Study 1, the developers and designers working closely together in Study 2 and, in Study 3, the developer and designers learning to work closely together while trialling Scrum.

Features of the integration work

In the analysis of accounts of practice, practitioner reports converge on three strategies for illustrating how Agile development was combined with UX design: merging

processes, inserting methods, techniques and tools into the development work and adapting methods, techniques and tools. The empirical studies contained considerable variation in how they addressed the combination of Agile development and UX design, each having its own perspective on what aspects of the combination require investigation and emphasis. Accounts in empirical studies agree that the combination of Agile development and UX design requires overcoming the challenges that their differences present, while practitioner accounts present methods, techniques and tools that aim to maintain focus and coordination between tasks.

The observations during the field work conducted for this dissertation, focused on the interactions between the developers and designers. In Study 1, analysing the work of integrating UX design with Agile development attended to the different rhythms of work between the developers and designers and how integration was achieved via phased work consisting of discrete tasks. The roles of articulation work, situated action and cooperative work were identified. Study 2 represents an important shift in the focus of the observations of practice and subsequent analysis. In Study 2, the focus shifted from process-oriented observations and analysis to talk in interactions between the developers and designer. The integration of Agile development and UX design in Study 2 relied on tacit knowledge shared among the developers and designer, with integration achieved via fluid role boundaries. The roles of articulation work and situated action were reinforced, however, cooperative work became less effective as a mechanism for explaining the integration work. In Study 3, the focus remained on talk in the interactions between the developer and designers, and integration achieved via the immediately reciprocated communications between the developer and the designers. As with Study 2, the roles of articulation work and situated action were reinforced, and cooperative work was de-emphasised.

Achieving integration

In the analysis of accounts of practice, practitioner accounts and empirical study accounts were both found to emphasise processes, tools and techniques. However, what the processes, tools and techniques were intended to achieve was different between the practitioner and empirical study accounts. Practitioner accounts emphasise processes, tools and techniques that achieve Agile development/UX design integration by establishing the right focus and coordination between the tasks of the developers and designers. Empirical study accounts emphasise processes, tools and techniques that achieve Agile development/UX design integration based on an understanding of the similarities and differences between them. The processes, tools and techniques in empirical study accounts aim to overcome the differences between Agile development and UX design.

The field work conducted for this dissertation, generated themes that showed how Agile development and UX design integration was negotiated and achieved in each setting, and that how integration was achieved depended in important ways on the work setting. Study 1 highlighted the role of expectations about behaviour, Study 2 the role of mutual awareness, Study 3 engaging and negotiating progress. Together with the findings from the literature, these themes are presented as facets of integration that contribute to an understanding of how integration is achieved in practice.

1.4 Contributions

This research presents an investigation into how Agile development and UX design are combined. Accounts of practice from the literature and data generated from field work with teams in organisational settings were analysed. The main contribution of this research is that **integration is an on-going achievement in practice.** Whereas previous work has focused on developing processes, tools and methods to

integrate Agile development with UX design, this research presents Agile development and UX design integration as negotiated on a day-to-day basis. This research draws attention to the work required to achieve integration and how that work depends on the settings in which the practitioners carry out their work. To achieve integration the following have to be enabled and maintained between the developers and designers on a day-to-day basis: *focus and coordination, mutual awareness, expectations about acceptable behaviour, negotiating progress* and *engaging with each other.*

This research explains how these facets of integration are **shaped by values embedded in the setting** in which practitioners work. That is, how focus and co-ordination, mutual awareness, expectations about acceptable behaviour, negotiating progress and engaging with each other manifests in each setting depends on the values endorsed by the organisations in which the developers and designers are embedded. These were values concerning *how best to develop quality software.* Two views emerge: (1) the best way of developing quality software is by keeping the Agile developers and UX designers separate, and (2) the best way of developing quality software is via the Agile developers and UX designers working closely together.

The **analysis of accounts of practice selected from published reports** identify the claims about combining Agile development with UX design on which the accounts converge. However, the **unresolved conflicts** within and between accounts contribute to a confused picture on how Agile development and UX design is combined in practice. The analysis highlights where the discourse regarding practice becomes disjointed: in the treatment of the differences' between Agile development and UX design, and the assumptions about the status of UX design and Agile development in organisations. Further, the accounts have little to say about how *work settings* shape the combination of Agile development and UX design in practice. Carrying out the research for this dissertation, therefore, required *engaging with practitioners in the settings in which they work.*

1.5 Road map

The outline for the rest of the dissertation is as follows:

Chapter 2 – Literature review presents an overview of how the combination of Agile development and UX design has been addressed in the literature, why the combination is a research problem of interest and what has shaped the discussions and approaches to combining Agile development and UX design. The chapter sets out what is implied in the combination and the debates surrounding Big Design Up Front. The tensions arising from practice as reported in the literature, opportunities for bridging the SE/HCI gap and the importance of setting for the combination is highlighted.

Chapter 3 – Research design presents the methodological foundation of this research and how the research questions were addressed. The research design is presented, setting out the approach to analysing the accounts of practice and the studies with teams in organisational settings. The limitations and ethical issues are considered.

Chapter 4 – Study 1 presents the findings from the study with Team1, reporting on features of the practical setting and features of the work of integrating Agile development with UX design. The main themes of the situated nature of Agile development and UX design work, dependencies, and expectations about behaviour are developed. This study highlights the importance of articulation work, cooperative work and situated action.

Chapter 5 – Study 2 presents the findings from the study with Team2A and Team2B, reporting on features of their practical setting and features of the work of integrating Agile development with UX design. The main themes of bridging roles and mutual awareness are developed. This study reinforces the notions of articulation work and situated action.

Chapter 6 – Study 3 presents the findings from the study with Team3, reporting on features of the practical setting and features of the work of integrating Agile

development with UX design. The main themes of engaging with each other and negotiating progress are developed. This study reinforces the notions of articulation work and situated action.

Chapter 7 – Accounts of practice presents the method and findings from the thematic analysis of the accounts of practice selected from the literature. The method is presented in three stages — comprising of two stages of searching and a thematic analysis stage. The grey and excluded literature are discussed. Considerations of the different types of evidence found in the literature are presented along with limitations of the analysis. The analysis examines the combination of Agile development and UX design as presented in accounts of practice — what is considered to be the problem, how it has been addressed and what is required for the combination to work.

Chapter 8 – Discussion presents the discussion that brings together the findings from the analysis of the accounts of practice and the findings from the field work. The shaping role of contextual values on the work of integrating Agile development with UX design is discussed. In practice, workflow and progress is maintained by maintaining focus and coordination between their tasks, expecting certain behaviours from others, maintaining mutual awareness, negotiating progress and engaging with each other.

Chapter 9 – Conclusion concludes the dissertation. The contributions of this research and avenues of future work are presented.

1.6 Publications

Parts of this research have been published elsewhere, as indicated:

Work in progress The early stages of the research were presented at the *Agile 2008* conference held in Toronto, Canada [Ferreira 2008].

Chapter 4 The findings from the ethnographically-informed study with Team1 have been published in a special issue of the journal *Software: Practice and Experience* [Fer-

reira et al. 2011].

Chapters 4 and 5 *Some of the findings from Study 1 and Study 2 were presented* at the *XP 2010* conference held in Trondheim, Norway [Ferreira et al. 2010].

Chapter 6 The findings from Study 3 contributed in part to an experience report presented at the *XP 2010* conference held in Trondheim, Norway [Tzanidou and Ferreira 2010].

Chapters 4–6 The findings from Study 1, Study 2 and Study 3 will be presented at the *Agile 2012* conference to be held in Dallas, TX [Ferreira et al. 2012].

CHAPTER 2

Literature review

2.1 Introduction

The need for better integration of Human-Computer Interaction (HCI) and Software Engineering (SE) has been recognised for many years. A review of the Agile/UX literature shows that integrating Agile development and UX design extends the HCI/SE discussion (§ 2.2). As with HCI and SE, attempts to combine Agile development and UX design involve fitting the activities from Agile development and UX design into the software development life cycle. If practitioners can find ways to work together, there are clear benefits for the practitioners, as well as for the quality of the software they deliver. However, as the review of the literature in this chapter, and a further analysis of accounts of practice in chapter 7 will show, the problems that practitioners have to work through in order to achieve these benefits, are not trivial (§ 2.3). The Agile/UX literature presents the differences between Agile developers and UX designers as competing and clashing. Yet, those who overcome their differences reap the rewards.

In this chapter, the literature is reviewed to find how the combination of Agile development and UX design has been addressed and what has shaped the discussion of the combination. The literature reviewed shows that there are problems that arise when Agile development is combined with UX design. However, no clear picture emerges for how the problems should be addressed. The review highlights features of the discussion on the combination of Agile development and UX design that prevent clarity and solutions from converging. The first feature of the discussion is that there are underlying debates around requirements and design (§ 2.4). The second feature is the variation in design approaches that are being combined with Agile development (§ 2.5). Published reports treat the types of design approaches (and Agile development approaches) as interchangeable, resulting in an assortment of underlying perspectives and assumptions that are never discussed. The third feature is that of setting (§ 2.6). The Agile/UX literature holds implications for what is claimed to be known about practice, specifically with respect to practical settings and how Agile development

and UX design unfolds in practical settings. Section 2.7 summarises this chapter.

2.2 Human-Computer Interaction and Software Engineering

The discussion around combining Agile development with UX design shares important features with the discussions on how Human-Computer Interaction (HCI) and Software Engineering (SE) can be integrated. Human-Computer Interaction (HCI) and Software Engineering (SE) have emerged as separate disciplines. Distinctions between the two disciplines have been made on the basis of having different vocabularies [Belenguer et al. 2003], different approaches to software development [Ferre 2003], even a lack of "sound scientific common ground," [Coutaz and Taylor 1995]. These differences are seen as obstacles to bringing HCI and SE together, that have accumulated over time [Grudin and Fielding 1995] and contributed to a gap between HCI and SE. The considerable literature on combining HCI and SE shows the ongoing efforts of researchers and practitioners to find common ground and bridge the gap between them.

Law [2003] examines the historical roots of HCI and SE and concludes that the differences are difficult to overcome, yet can be narrowed through dialogue between the two disciplines. Folmer et al. [2006] present extensions to interaction design patterns, called bridging patterns, such that architectural impacts of usability improvements on the software under development can be assessed. Constantine et al. [2003] illustrate the connection that Usage-Centered Design [Constantine and Lockwood 1999] has with software engineering, arguing that use cases (as described by McMenamin and Palmer [1984]) are the common connection. Their paper calls for tools that improve on the UML's (as described by Rumbaugh et al. [1999]) lack of constructs for designing a user interface, and that support the interconnections between HCI and SE. A similar approach is taken by Paternò [2001], who integrate UML with another notation (ConcurTaskTrees as described by Paternò [1999]) in order to extend the capabilities of the notation to modeling user interfaces.

Another route to bridging HCI and SE considers fitting both into the software development life cycle as key. This notion is captured by Seffah and Metzker [2004]'s question:

> "The obvious question is: where should UCD techniques and knowledge be considered in the existing software development life cycle to maximize benefits gained from both SE and UCD?"

Ferre et al. [2005] map HCI activities to SE activities as the basis for developing a framework such that developers may be aware of the appropriate times in the life cycle for applying HCI techniques. In developing an application for the mobile Juárez-Ramírez et al. [2011] integrate best practices from software engineering, usability engineering and human-computer interaction. Haesen et al. [2008] combine principles and practices from software engineering and user-centered design in a process framework they call MuiCSer. Searching the literature brings up many more examples (e.g. the book edited by Seffah et al. [2005] contains further examples).

Although the above is far from an exhaustive review of the area, the importance of tools, techniques and methods, and how they fit into the software development life cycle, is clearly a concern in bridging HCI and SE. The emphasis on tools is recognised in a systematic literature review conducted by Bjørnson and Dingsøyr [2008]. This indicates that closing the gap between HCI and SE lies in the tools, techniques and methods that HCI and SE practitioners employ.

A number of researchers have proposed that the combination of Agile development with UX design brings opportunities for bridging the gap between HCI and SE (e.g. Sharp et al. [2004] propose that this may be achieved by combining XP and UCD) and study the combination in this context [Chamberlain et al. 2006; Hussain et al. 2009b; Fox et al. 2008; Memmel et al. 2007a]. The following examples show how the UX design approach has been found to support Agile development and vice versa.

Up-front UX design supports building and maintaining Agile development's notion of "vision" by providing a holistic view of the software under development [Ferreira

2007; McInerney and Maurer 2005; Wilcox et al. 2007]. The visual editing tools employed in UX design, allow designers to express conceptual notions about the software in some physical form early in development [Lievesley and Yee 2006]. In the case of Obendorf and Finck [2008], the scenarios from UX design enabled developers to expose important assumptions before the designs were turned into working software. The scenarios and personas of UX design have been found to support Agile developers' interactions with other non-developer stakeholders. For example, Obendorf and Finck [2008] report that scenarios were used to "establish a common understanding of the work task with the involved stakeholders." They describe how the simple language of a scenario aided the communication between developers and end users, who had been struggling with the software engineering terminology. Scenarios and personas have also been found to support prioritisation of features by providing Agile developers with a better understanding of the context in which end users will use the software [Chamberlain et al. 2006; Obendorf and Finck 2008; Patton 2002a; Ungar and White 2008]. The focal roles from Usage-Centered Design aided prioritisation of features and helped developers to make decisions about which features to implement [Patton 2003].

In return, Agile development supports UX design with opportunities for obtaining and incorporating feedback at regular intervals. Miller [2005] reports that the quantity of end user feedback and the impact that the feedback was seen to have on subsequent development was higher with Agile development than what had been the case before adopting Agile development. Armitage [2004] notes that Agile development allowed end user feedback to influence subsequent development earlier in the process than with traditional software development. Implementing UX designs during the Agile iterations, was found to bring insights into design and allow for the refinement of design details [Ferreira et al. 2007b]. Agile developers, with their knowledge and experience of software, provided useful feedback on initial UX designs by exposing gaps or incomplete information [Ohlhauser 2008]. Agile development provides opportunities for UX designers to be involved throughout the development

effort, have an impact on design decisions throughout [Ferreira et al. 2007c] and direct the focus of development on the user [Najafi and Toyoshiba 2008]. Armitage [2004] credits Agile development's "lower risk release cycles" with encouraging design experiments with actual end users, which agrees with advice by Bill Buxton, an expert UX designer: "We ideally need to be able to experience our designs in the wild during the early stages of the process ...the earlier the better" [Buxton 2007, p 37]. While the discussion in this section displays clear benefits for combining Agile development and UX design, as further review (and the analysis of accounts of practice in chapter 7) will show, the problems that practitioners have to work through in order to achieve these benefits, are not trivial.

2.3 Tensions that combining Agile development and UX design brings for practice

How Agile development and UX design are combined in practice is addressed in detail in chapter 7, which focuses on accounts of practice. This section focuses on the reported challenges that the combination presents for practitioners. Most accounts of practice are positive, yet consideration is given to significant tensions that arise from the realities of practice. Practitioners have reported on the various ways in which they deal with the challenges. Some authors report on techniques and practices that directly address these tensions, while others suggest new and adapted methods (e.g. Beyer et al. [2004], Lee and McCrickard [2007] and Constantine [2002]). Singh [2008] introduced a new role, that of usability product owner, into the Scrum process.

Tensions emerging from reports on Agile development and UX design stem not only from the types of activities required during the development effort, but also from the people involved in the development effort. Tensions as a result of the development activities include:

- The timing/scheduling of Agile practices and UX design techniques;

25

- Differing approaches to software implementation.

Tensions as a result of the people involved in the development effort include:

- Power struggles;

- Cultural differences.

Social tensions such as power struggles and cultural differences are not unique to combining UX design with Agile development, however they enjoy visibility in the accounts of Agile development and UX design in practice that draws attention to the differences between the Agile development and UX design disciplines.

2.3.1   Timing/scheduling

Timing/scheduling tensions of UX design activities and Agile development activities found in reports on their combination, can be attributed to specific characteristics of Agile development methods. The main characteristics underlying the tensions highlighted in this section relate to (1) the development effort segmented into iterations; (2) the functional focus of Agile testing. The literature shows that these characteristics make it unclear how users are to be involved and how their feedback can be incorporated back into the development effort, how UX designers and Agile developers can coordinate their activities with each other and with other non-Agile teams and when usability testing can be performed in the context of other Agile development tests, e.g. unit testing and acceptance testing.

With the Agile development effort segmented into iterations, practitioners are unclear as to when end users should become involved in the process [Detweiler 2007; Wolkerstorfer et al. 2008] and when user feedback can be integrated back into the development effort [Chamberlain et al. 2006; Lee and McCrickard 2007; Detweiler 2007; McInerney and Maurer 2005; Wolkerstorfer et al. 2008]. UX design practitioners may feel that Agile development does not present enough opportunities for obtaining user feedback [Sy 2007], while there is evidence for practitioners seeing Agile iterations

and release cycles as appropriate points in the development effort to perform usability testing [Ferreira et al. 2007a]. Armitage [2004] reports that feedback from regular releases of the product can serve the same purpose as organised usability testing. This was also the case for Wilcox et al. [2007]. Najafi and Toyoshiba [2008] report that the first week of each Agile iteration in their development effort was dedicated to user research and testing implemented features. McInerney and Maurer [2005] report that one of their participants devoted two weeks in the release cycle to fixing usability problems highlighted by usability testing. Ohlhauser [2008] reports receiving usability feedback from their customer after the first release and dedicated one Agile iteration after the release to incorporating customer feedback. Hussain et al. [2008a] present an alternative approach to the scheduling of end-user testing, by performing end-user testing only once the customer's business knowledge indicates that testing the application will be effective from the business perspective. Gaining repeated access to the right end users can be problematic [Federoff et al. 2008], and Ohlhauser [2008] reports using individuals from the organisation for usability testing.

Coordinating UX design activities and Agile development activities poses a challenge. Hakim et al. [2003] present a tool called "Sprint" to help designers "stay synchronised with customers, analysts, and developers". This tool electronically links required functionality for the software across various development artefacts such as personas, scenarios and use cases. During the time the Agile developers implement the code, questions affecting the UX design may conveniently be resolved if UX designers are on hand during the development process [Ferreira et al. 2007a; Wolkerstorfer et al. 2008]. Ferreira [2007] has pointed out a possible weakness of this strategy: that developers may not always approach the UX designers right at the moment when it is required and then forget to do so later. Coordinating development work with other non-Agile teams was also reported to be problematic by Sy and Miller [2008], especially when user experience work is divided among separate teams. Miller [2005] reports that a possible solution to this would be to have the UX designers responsible for the whole user experience.

The popular view from reports of practice seems to favour adapting UX design techniques for a better fit with an Agile approach to development (e.g. [Detweiler 2007; Federoff et al. 2008]). There are reports on how UX design techniques can be adapted to be more light weight, in order to fit with the short Agile iterations [Beyer et al. 2004; Constantine 2002; Sy 2007]. There are also reports that present a view on the UX designer role, guidance targeted at changing the way designers are accustomed to working. Armitage [2004] presents eight guidelines for designers working in an Agile environment — designers are encouraged to be content with producing partial solutions, designing for change later in the development effort and designing the simplest possible solution. Lievesley and Yee [2006] found that UX designers may not be accustomed to working closely with developers, that their holistic view should remain "malleable" and "the designer rules out as little as possible until as late as possible." Ungar and White [2008] point out that "the user-centered designer working within Agile needs to adapt quickly as perceived business value changes."

Reports in the literature suggest extending or re-focusing the established tests of Agile methods to involve usability testing. Wolkerstorfer et al. [2008] suggest extending unit tests to test 'interaction flows', which integrates usability testing with the test-first strategy of Agile development. Lee and McCrickard [2007] suggest performing lightweight usability testing as part of the acceptance testing process. This approach is successful when usability tests are smaller, more focused and performed more often [Lee and McCrickard 2007].

2.3.2 Implementation approaches

The tensions found in the literature between UX design and Agile development, due to their different approaches to implementation, can be attributed to the following specific characteristics of Agile practice: (1) the Agile preference for minimal design before implementation; (2) welcoming changing requirements during implementation; and (3) decomposing and estimating the product in terms of small implementable

tasks. The literature highlighted in this section shows that these characteristics can challenge the development of a holistic view of the product and affect UX designers, who aim to maintain consistency in their designs.

UX designers are accustomed to extensive user research and design before implementation begins, while Agile development advocates starting the coding without up-front design. This forms one of the fundamental debates for the combination of Agile development and UX design (§ 2.4). Although Agile values discourage up-front planning activities for software design, i.e., up-front design of the code [Fowler 2004], this attitude can be seen to affect time allocated for UX design on Agile projects. UX designers working with Agile development teams find that time for up-front UX design and user research is short [Ambler 2008; Chamberlain et al. 2006; Detweiler 2007; Sy and Miller 2008; Ungar and White 2008], and in some cases there is no allocated time for up-front research [Sy 2007]. Practitioners have been found to do some UX design up front in order to get the development effort started [Ferreira et al. 2007b; Fox et al. 2008; Hodgetts 2005] and in fact consider up-front UX design more acceptable than up-front code design [Ferreira et al. 2007a]. Further, practitioners credited the up-front design activities with helping to mitigate problems and allowing early customer input [Ferreira et al. 2007a]. Miller [2005] and Sy [2007] explicitly call this stage before implementation begins, "Cycle Zero," also referred to in practice as "Sprint Zero" [Najafi and Toyoshiba 2008].

In order to fit with the shorter timescales of Agile development, Sy [2007] suggests adjusting the timing and granularity of user research. The literature offers little guidance as to how much UX design is appropriate *during* Sprint Zero. Constantine [2002] proposes that work before development begins is kept to a minimum, establishing the following: (1) an overall organisation of the different parts of the UI (2) a versatile navigation scheme (3) a consistent look-and-feel for user tasks. The literature offers little guidance as to how much UX design is appropriate *after* Sprint Zero and once the Agile development iterations begin. Sy and Miller [2008] advise that UX designers work one iteration ahead of the Agile developers. However, Ungar and White [2008],

29

recommend completing enough up-front design for two to three Agile iterations worth of implementation.

Agile development embraces changing requirements throughout the course of development with iterative and incremental development [Beck 1999], whereas UX design traditionally deals with one set of requirements which is iteratively refined during the design stage, but not during implementation. Haikara [2007] reports on this tension arising between the UX designers and Agile developers. The UX designers create personas drawn from a single, one-off set of requirements, while Agile development welcomes new requirements throughout implementation. Further, UX designers expected to deliver to the end user once, after refining their designs with the end user, to help ensure of the consistency of their designs [Haikara 2007]. Along with the frequent releases of Agile development and the iterative and incremental approach, UX designers may have difficulty maintaining consistency in their designs [Lee and McCrickard 2007].

When Agile projects allocate little time for performing UX design before implementation begins, and allow requirements to change during development, the holistic view of the product is compromised and in some instances seen to be missing [Constantine 2002; Lee and McCrickard 2007; Sy 2007; Wolkerstorfer et al. 2008]. Agile development relies on developers decomposing tasks into small implementable parts and then estimating the amount of time it would take to implement those tasks, but it is not clear how UX design work can be similarly decomposed and implemented. Armitage [2004] relates that UX design was decomposed into smaller implementable features, but gives no details of how this was achieved. In practice, planning detailed UX design tasks within iterations is troublesome [Ferreira 2008; Hodgetts 2005], suggesting that some UX design tasks are "highly creative, even artistic in nature" and therefore, difficult to estimate and plan. In the account by Hodgetts [2005], estimations for these highly creative tasks were "...time box[ed] ...with a fixed estimate representing the most time they would want to waste if this task could not be completed as envisioned".

2.3.3 Power struggles and cultural differences

A general lack of understanding about the role of UX design in Agile development (§ 2.4.2), compounded by the pragmatic values of Agile development, leaves UX designers struggling for power and understanding of their work. While developers can be secure in their knowledge that if they are sent home then "there is no system" [Highsmith 2002, p 47], UX designers, who do not participate in the programming activities, can be less secure. There is a sense in the user experience community that Agile methods neglect the user experience [Beyer et al. 2004; Constantine 2002; Lee 2006]. Armitage [2004], a user experience expert claims: "...the Agile community rarely mentions users or user interfaces at all, which means that either they neglect the user experience or are focusing on projects with less need for sophistication in user experience". Hodgetts [2005] concludes that this absence results in a disenfranchised user experience community. McInerney and Maurer [2005] attribute this "lack" of consideration of UX design issues to the UX designer not being recognised as a core role on the team. Beck and Andres [2004] identified a UX designer role as necessary on the Agile team, but direction on how this role is integrated into the Agile team was limited.

In the account by Broschinsky and Baker [2008], the organisation formed the XP team first and retrospectively added UX designers. The UX designers were therefore expected to integrate with the work of the Agile developers. Similarly, in the case of Düchting et al. [2007], usability was considered as just another attribute of software that could be "added" to the development process. Lievesley and Yee [2006] report that UX designers joined a Scrum team two to three weeks after the development effort had begun. Therefore there is evidence for the UX designers not being part of the decision to use Agile development methods, as in the case of the developers transitioning to Agile development in the account by Budwig et al. [2009].

The lack of understanding of UX design in the Agile development context, may require that UX designers defend and justify their designs [Lee and McCrickard 2007;

McInerney and Maurer 2005; Ungar and White 2008]. Broschinsky and Baker [2008] explain that the role of the UX designer on the Agile team was unclear to the Agile developers, and to gain recognition from the Agile team the person in that role had to make themselves visible and heard. The same authors describe how the UX designer had to satisfy the developers that the personas they had created were accurate by introducing the developers to real customers on which the personas were based [Broschinsky and Baker 2008]. They were also careful to refer to "the data" when communicating about their work to avoid the developers assuming the designs were based on the opinions of the designers [Broschinsky and Baker 2008]. Detweiler [2007] reports a similar experience.

Beck [1999] asks "What is the simplest thing that could possibly work?" in order to encourage Agile teams to remove unnecessary complexity and concentrate on producing working software that adheres to today's requirements. Lievesley and Yee [2006] cited this principle when they found developers implementing software that deviated from the specifications developed by the UX designers. In practice, Agile developers find themselves making implementation decisions when UX designers are not on hand to answer their questions [Ferreira 2007]. Unfortunately, their decisions could result in software that does not correspond directly to the UX designer's specifications and potentially compromise the resulting user experience. Singh [2008] notes that this principle encourages a focus on functionality and results in lower priority for usability issues. Najafi and Toyoshiba [2008] compare two projects to show the impact of co-location on Agile projects. On one project the UX designers were in the same location as the Agile development team, whereas on the other team they were in different locations (distributed). The distributed team lagged behind the release schedule and experienced more significant usability problems than the co-located team. UX designers want to have an influence on the prioritisation of features [Detweiler 2007; Sy 2007] to ensure that the priorities of features reflect user needs [Federoff et al. 2008]. Sy [2007] relates an instance of where the UX designers were ignored during the prioritisation activities.

Hodgetts [2005] calls for understanding and respect between all disciplines on Agile teams, in order to foster collaboration. Ungar and White [2008] propose that a technique known as the "design studio" can foster collaboration between UX designers and Agile developers. Miller [2005] reports on how developers and UX designers were collaborating on a daily basis in order to organise their work. Chamberlain et al. [2006] and Detweiler [2007] favour closer collaboration between UX designers and Agile developers.

2.4 Debates that shape the discussion on combining Agile development and UX design

This section outlines relevant debates that shape the discussion on the combination of Agile development with UX design. The combination of Agile development with UX design intersects with debates concerning software design and construction, and with views on the nature of requirements and design. Understanding how Agile development is combined with UX design also requires an understanding of the various positions driving the debates.

2.4.1 Big Design Up Front

The Portland Pattern Repository[1] describes BDUF: "The term Big Design Up Front is commonly used to describe methods of software development where a 'big' design is created before coding and testing takes place." According to accounts in the literature, BDUF is something Agile developers avoid [Beyer et al. 2004], is contrary to Agile development principles and values [Ferreira 2007] and refers to traditional, non-Agile software development methodologies [Lindvall et al. 2002].

In an Agile world, anything that fixes requirements up front and constrains ongoing change is viewed with caution. This polarises views on how Agile development

---

[1]http://c2.com/cgi/wiki?BigDesignUpFront

can be combined with UX design because plan-driven disciplines, that traditionally carry out work before coding begins, is on uncertain ground with respect to Agile development. One such discipline that carries out work before coding begins is UX design (software architecture another [Nord and Tomayko 2006]). In the literature, conflicting positions on the nature of *requirements* and *design* drive the discussion on how UX design can be combined with Agile development.

Requirements

Constituencies hold differing assumptions about requirements — particularly assumptions about the point at which requirements can be known. On one side, there are those that attribute the fast pace of change in the industry and in technology to an inability to anticipate requirements up front: "The industry and the technology move too fast and customers have become increasingly unable to definitively state their needs up front" [Lindvall et al. 2002]. On the other, there is a view that requirements are knowable up front, as Highsmith and Cockburn [2001] explain: "Traditional approaches assumed that if we just tried hard enough, we could anticipate the complete set of requirements early and reduce cost by eliminating change."

In the debate between Kent Beck and Alan Cooper, both agreed that customers change their minds about what they want midway through projects [Nelson 2002]. For Beck, this is part of the reality of software development that XP is designed to deal with. For Cooper, the changing requirements are a result of customers not understanding their requirements. Therefore, Cooper views interaction designers as essential to alleviating this "problem" for the Agile developers by determining the requirements up front [Cooper 2010]. In the debate surrounding BDUF, views on design, and whether design is seen as emergent or fixed, orient to the assumptions about whether or not requirements can be anticipated up front.

Design

Nerur and Balijepally [2007] contrast two types of design on the basis of their approach to problem-solving:

1. **Traditional design** characterised by "Selection of best means to accomplish a given end through well-planned, formalized activities."

2. **Emergent design** characterised by "Learning through experimentation and introspection, constantly reframing the problem and its solution."

In "The New Methodology", Fowler [2005] explains that plan-driven design (i.e. planning before you build) aims to turn software construction into a predictable activity and that this requires separating design from construction. In his article "Is Design Dead?" Fowler [2004] explains: "Once the design is done they can hand it off to a separate group (or even a separate company) to build [...] The programmers can follow the direction of the design and, providing they follow the design, have a well built system." He contrasts this with evolutionary design (which parallels the emergent design of Nerur and Balijepally [2007]): "Design is part of the programming processes and as the program evolves the design changes." Kent Beck and Alan Cooper famously debated the place of UX design in XP [Nelson 2002]. While Cooper advocated UX design being done entirely before any implementation, Beck argued for a phaseless approach in which implementation work is as cheap as UX design prototypes, in terms of effort and time.

In the context of combining UX design and Agile development, there are accounts that uphold the separation between design and construction on the basis that emergent design compromises the user experience. Constantine [2002] is one advocate for up-front design: "... some minimum up-front design is needed for the user interface to be well-organised and to present users with a consistent and comprehensible interface." Meszaros and Aston [2006] agree that "Emergent Design doesn't work very well for user interfaces." and propose that "Some Design Up Front seems to provide

35

better guidance to the development team and provides earlier opportunities for feedback." However, up-front design, and how much is enough, remains controversial in the Agile/UX literature [Lee 2006; Ferreira et al. 2007b; Adikari et al. 2009].

2.4.2 Place of design and construction

Aside from when the user experience should be designed in relation to coding, there has been little discussion around the question about what is seen as generating value. In the literature, there is evidence that value can occur without the UX designers — that UX designers are only brought in when Agile development is found to not be 'enough'. In the report by Broschinsky and Baker [2008], the Human Factors specialists came on board once the XP team realised they were still "missing" something and "to overcome the loss of cash flow by focusing on customer needs." From the account, it is clear that the XP team had existed and developed software without the Human Factors specialists. Patton [2002a,b] claims that the XP team are delivering value, however, improving on that required the addition of Usage-Centered Design. Kollmann et al. [2009] report that design can be compromised in favour of delivery of the software.

That is, UX designers are seen to add qualities to the software that are desirable but not essential, which Seffah and Metzker [2004] have noted as presenting an obstacle to integrating user-centered design and software engineering. This trend has been recognised in the wider HCI/SE community [Göransson et al. 2003]. Coutaz and Taylor [1995, p. 2], for example, report on the results of a workshop held with HCI and SE researchers, in which the view that HCI is considered an expensive addition and requires specialists appeared frequently among participants. Although this point has not been discussed in detail in the context of UX design and Agile development, this is significant because it shows that the relationship between Agile development and design is not symmetric and complementary, despite being portrayed as such. The analysis of accounts of practice in this dissertation will bring up some of the issues

around integrating the two processes as if they are on equal footing (see chapter 7). There is evidence that designers report on the extra work they do in order to establish their value for software development (§ 7.6.4).

2.5 *Design and Agile development: What is being combined?

Section 1.1 set out the reasons for choosing the terms *User Experience Design* and *Agile Development* for this dissertation. This section reviews the Agile/UX literature to find first which *design terms are used for the approach for involving and maintaining focus on the end user in software development. Next, the Agile development terms used in the combination are discussed, including what the choices of terms may entail. It is necessary to take a closer look at what is being combined, and what constitutes the Agile/UX literature, as (1) the current literature treats the types of design approaches (and Agile development approaches) as interchangeable, and (2) there is a need for a better understanding of the assortment of perspectives and assumptions underlying discussions about combining Agile development and UX design. The aim here is not to assign definitions to the terms used in the literature, but to highlight the assumptions and the variety that the literature accepts.

2.5.1 *design terms found in the literature

Table 2.1 shows the profusion of terms appearing in the Agile/UX literature. These terms represent the design approach named in accounts in the literature and as it is used in combination with some form of Agile development. The reported reasons for choosing a particular type of design, as found in the accounts referenced in table 2.1, provide a snapshot of the seemingly arbitrary nature in which these combinations make their appearance in the literature.

In the practitioner reports by Patton [2002a,b] and Hodgetts [2005] Usage-Centered

Design and User Experience Design are *discovered* by the respective authors:

> "The UED practitioners I met helpfully directed me to the works of Larry
> Constantine [Constantine and Lockwood 1999] and Alan Cooper [Cooper
> and Reimann 2003]. The most helpful to me, however, was a relatively
> small, but significant book by Jesse James Garrett [Garrett 2002]," [Hodgetts 2005].

In their accounts, Patton and Hodgetts are both referring to Constantine and Lockwood [1999], the creators of Usage-Centered Design. Hodgetts mentions an awareness of the work of Cooper and Reimann [2003], but favours "user experience design" as described by Garrett [2002] and gives an account of coaching Agile teams trying to include user experience design. For others, such as Miller [2005] and Broschinsky and Baker [2008], the design approaches in their accounts are those that already exist in the organisations they are part of. Broschinsky and Baker [2008] explain how the authors met, by referencing an existing Human Factors team in the organisation:

> "David Broschinsky and Lisa Baker met in April 2003, when Lisa hired
> Dave for LANDesk Software in an effort to start rebuilding their Human
> Factors team."

The account by Broschinsky and Baker [2008] contains no explanation of where they take the term "Human Factors" from, although they emphasise the use of personas, which they associate with Alan Cooper (no reference). Miller [2005] references Norman and Draper [1986] with respect to "User-Centered Design" and lists some methods that could be used as part of a User-Centered Design approach, e.g. Focus groups [Greenbaum 2000] and Contextual design [Beyer and Holtzblatt 1997].

Beyer et al. [2004] present an adapted form of "Contextual Design" [Beyer and Holtzblatt 1997], which they call "Rapid Contextual Design":

> "Contextual Design is a well-respected user-centered design method that
> has been around for over 10 years." and "CD can coexist with agile

methods . . . the two complement each other so well that they form a very strong combination."

The UX team in the account by Cho [2009] adopt Goal-Directed Design [Cooper et al. 2007], due to the team's past successes with this approach. Kane [2003] chooses to

"challenge the agile development community to find ways to incorporate and gain value from discount usability engineering practices."

Kane [2003] uses "discount usability engineering" as described by Nielsen [1989] and gives no further clues as to the reasons for choosing discount usability engineering. Sohaib and Khan [2010] refer to "usability engineering." While their review includes accounts that use a range of design approaches, the authors are in effect relabeling that collection of approaches as "usability engineering" without explicitly acknowledging this.

Researchers interested in the combination of a design approach with Agile development, report on what they consider to be *established* design approaches, as Lee and McCrickard [2007] do when reporting on Scenario-Based Design:

"We concentrate on collaboration between interaction designers and developers because both are strongly involved in many kinds of successful software," Brown et al. [2008].

However, it is not always explicitly stated whether researchers are reporting on the type of design they encounter in practice, or whether they are relabeling for the purposes of their discussion. An exception is the report by Kollmann et al. [2009], in which the collective "user experience designer" perspective is constructed from interviews with roles named "UX architect", "UX consultant", "information architect" and "interaction designer".

Table 2.1: *Design terms found in the Agile/UX literature

interaction design	[Patton 2002a]
usage centered design	[Patton 2002b]
discount usability engineering	[Kane 2003]
Rapid Contextual Design	[Beyer et al. 2004]
user experience design	[Hodgetts 2005]
user centered design	[Miller 2005]
scenario-based design	[Lee and McCrickard 2007]
human factors	[Broschinsky and Baker 2008]
goal-directed design	[Cho 2009]
usability engineering	[Sohaib and Khan 2010]

There is little consensus, in practice or in the wider design literature, about the boundaries between approaches and roles. For example, Saffer [2010, p. 3] explains that interaction design as a discipline is difficult to define partly due to its "multidisciplinary roots." As another example, Gulliksen et al. [2008] claim that the difficulty in measuring user experience lies in the disagreement about what user experience is. "The consequence of the diversity on what UCD really stands for makes it in practice a vague concept that can be interpreted in many ways" [Gulliksen et al. 2003]. This confusion carries over into the Agile/UX literature and becomes evident in the roles associated with the type of design they are reported to be doing. Miller [2005] reports on their organisation having a "usability engineering team" made up of interaction designers and graphic designers, doing user-centered design. Like Miller [2005], Lievesley and Yee [2006] describe interaction designers doing user-centered design, while in Patton [2002a] interaction design is usage-centred design and in Cho [2009] the "User Experience designers" adopted "Cooper's Goal-Directed design (GDD) [Cooper et al. 2007] methodology."

From the descriptions of *design teams in the Agile/UX literature it also becomes clear that *design in practice is carried out by different roles. Miller [2005] includes this observation in their account:

> "It also seems common for companies to assign the various UCD duties to separate groups, one group doing the market research and gathering user requirements, another doing interface design, and a third doing usability testing."

How the disciplines and roles mentioned in the Agile/UX literature fit together is left implicit. To take Miller [2005] as an example again[2], they mention that their interaction designers are responsible for the things that other organisations split up into different roles, but omit explaining the differences between the graphic designers and interaction designers on the team.

It is possible to find definitions for each of these *design terms in the literature, however, the point of this discussion is to recognise the variation in terms and uses of those terms in the literature. The terms are not equivalent and not interchangeable, and they are used for a specific (albeit sometimes unstated) purpose in each report. It is reasonable to assume (and it is the view taken in this dissertation) that practitioner reports make use of the terms in a way that is particular to their teams or organisations, and there are scattered clues in the literature that practitioners recognise the situated nature of their approach. For example, Patton [2002b] acknowledges that what they call Usage-Centered Design will not match "the original text that describes it." Thus, design terms in practitioner reports take their meanings from the organisational settings in which they are embedded.

---

[2]Miller [2005] is frequently referred to in this discussion as this report is very well known in the Agile/UX field and considered by experts to be a successful example of the combination of Agile development and user-centered design.

2.5.2 Agile terms found in the literature

Table 2.2 illustrates the range of Agile terms appearing in the literature. These terms represent the Agile development approach being combined with some form of *design. As was done with the *design terms in section 2.5.1, the reports in table 2.2 provide various reasons for a particular type of Agile approach appearing in the combination with *design, and is representative of the reasons found in the Agile/UX literature.

In this set of reports, there is less evidence for a "discovery" of Agile development in the way that there is evidence of discovery of *design in the reports in table 2.1. Instead, we find practitioners citing reasons that indicate organisation-related reasons for moving to Agile development. In a report by Krohn et al. [2009], Feature-Driven Development (FDD) was a considered choice — it was a software development process that supported a "sorted list of feature sets" that emerged from applying UCD. Therefore, it was chosen as it was a good match for UCD, however, the reasons are not elaborated further. Wilcox et al. [2007] report on their in-house Agile development method that suited their needs:

> "Our development process borrows freely from the Agile family methodologies [Cockburn 2002]. We tailored our practices to take advantage of how hosted applications are deployed and used."

Miller [2005] reports on an event that caused the combination of Agile development with *design — that of the organisation transitioning to Agile development. In doing so, they work with an Agile guru to teach them about Agile development:

> "We brought in Jim Highsmith to teach agile principles and methodologies. The product development group chose to adopt the Adaptive Software Development [Highsmith 2000] process along with Scrum meetings [Schwaber and Beedle 2002] and many elements of Extreme Programming [Beck 1999]."

Other examples in the Agile/UX literature of these common occurrences in practice, include the report by Frank and Hartel [2009], who had Jeff Patton help with their process, and the report by Federoff and Courage [2009] who adopted Scrum when their organisation transitioned to Agile development.

Similar to the way the literature treats *design as something already existing within an organisation, the same can be said of the Agile development approach. Patton [2002a] is one such example and reports that although XP was delivering some value, there was still room for improvement:

> "In practice we found XP did deliver high quality software quickly, but
> the resulting product still failed to delight the customer."

For researchers the momentum with which Agile development is being taken up in industry is remarkable and therefore worthy of study: "The agile approach is quickly becoming mainstream in the software industry" [McInerney and Maurer 2005]. Hussain et al. [2009b] also note this:

> "Agile software development methods are quite popular nowadays and are
> being adopted at an increasing rate in the industry every year."

In both these reports, "Agile development" is being used as an umbrella term for the approaches that adhere to the values and principles of the Agile manifesto[3]. The final report to mention from table 2.2, is that of Haikara [2007]. This report is of an approach in which XP practices have been adapted for mobile development — Mobile-D, a product of Agile development research (see Abrahamsson et al. [2004] for more details on Mobile-D). However Haikara [2007] reports no reasons for choosing Mobile-D.

---

[3]agilemanifesto.org

Table 2.2: Agile terms found in the Agile/UX literature

eXtreme Programming (XP)	[Patton 2002a]
Agile as umbrella	[McInerney and Maurer 2005]
Agile development as a mix of XP, Scrum, and Adaptive Software Development	[Miller 2005]
Mobile-D	[Haikara 2007]
In-house agile development method	[Wilcox et al. 2007]
Scrum	[Federoff and Courage 2009]
Feature-Driven Development	[Krohn et al. 2009]

In the literature, there is not the same level of confusion around Agile development methods and what the terms mean, as there is around the *design terms. Scrum developers are not sometimes reported to be doing XP and XP developers are not sometimes reported to be doing Scrum. Also, when the umbrella term 'Agile' appears in the literature it presents a more coherent understanding of the collection of Agile methods than the disputed umbrella terms do for *design. This is a comment on the *literature* and how Agile development and *design is presented in the literature, it is not to say that practice presents any more or less confusion than the literature, or that Agile development is more or less understood than UX design. The literature does not address every detail of practice and much of what is found in the literature are convenient, but also necessary, conflations of the work that Agile developers and UX designers do.

2.5.3 Developers and designers are not the same

The Agile/UX literature portrays UX designers and Agile developers as distinct, even clashing groups of practitioners. As noted by Memmel et al. [2007b]: "HCI and SE

are recognized as professions made up of very distinct populations." These groups are assigned their own concerns, aims and approaches to creating software, which becomes part of the challenge of combining the HCI-related design disciplines with Agile development. On the one hand the Agile developers are portrayed as unconcerned with anything outside of coding: "software developers focus on functional aspects and neglect the non-functional characteristics of software like usability" [Sohaib and Khan 2011]. On the other, UX designers are portrayed as unwittingly thrust into interactions with a strange band of people. The account by Budwig et al. [2009] is an example where the UX designers are not involved in the organisation's decision to transition to Agile development and Lukanuski et al. [2008] found that UX practitioners are concerned that "Engineering, not User Experience/UCD drives the decision on taking an Agile/Scrum approach to a project." Based on a review of the Agile/UX literature, da Silva et al. [2011] insist that the outlook is positive despite this difference between the groups, and that improvements are to be had if they can find ways to work together:

> "Agile Methods have a distinct culture that at first glance seems to conflict with User-Centered Design (UCD) [McInerney and Maurer 2005]. However, according to these same authors, the use of agile methods can result in improved usability."

It is understood from the Agile/UX literature, that supporting the interests of the end user requires different tools, techniques, etc. than coding software. Developers are considered to "rarely be usability experts" [Chamberlain et al. 2006], so that UX designers are seen to bring specialised and valuable skills to Agile development teams [Ferreira 2007]. However, it is not always the case that these populations are as distinct in practice as they are portrayed by some in the literature. Studies of practice have found that those who carry out the UX design work may be Agile developers with an interest or training in UX design [Ferreira 2007; Fox et al. 2008]. Wolkerstorfer et al. [2008] and Patton [2002a] describe how combining these different

skills, by way of combining *design and Agile development, achieved various aims for their software development teams:

> "the advantages of Extreme Programming methodology (on-time delivering, optimised resource investments, short release cycles, working high quality software, tight customer integration) with the advantages of a user-centered design process (usable, accessible and accepted products, end-user integration)" [Wolkerstorfer et al. 2008].

> "Agile development methods allowed us to deliver high quality software sooner, and interaction design concepts lent us the degree of end-user empathy we were missing to help increase confidence that we hit our target of end-user satisfaction" [Patton 2002a].

It has to be noted that the Agile development approaches DSDM (Dynamic Systems Development Method) [Stapleton 1997] and Crystal Clear [Cockburn 2004] do take account of UX design, but set up the developer and designer roles in different ways to how this is done in the rest of the Agile/UX literature. DSDM includes the end user as a full-time role on the team, guiding the decisions of the development team. Since the finding by Abrahamsson et al. [2002] that there was no identifiable research on DSDM at the time, the reports by Schulze et al. [2005] and Hope and Amdahl [2011] have been published. Schulze et al. [2005] present a case study in which the challenge of user participation in the development process is addressed. There is little on the interactions between designers and developers, which is the focus of this dissertation. When the term "designer" appears in the account by Hope and Amdahl [2011], it is being used to refer to "systems designers" who "wrote code, and prepared for system functionality to make the programme work." Crystal Clear, includes an expert user on the team, and the user experience is designed in concert with the designer-programmer, business expert and expert user [Cockburn 2004, p. 174]. Other than Cockburn's accounts, no other accounts of experience with this methodology have been identified — similar to the state of affairs reported by Abrahamsson

et al. [2002].

What this section shows, is that the Agile/UX literature presents the differences between Agile developers and UX designers as competing and clashing. Yet, working together has advantages for teams who can overcome the differences. The problems that are introduced by the differences between the developers and designers is part of the discussion on bridging the HCI/SE gap, and is analysed in more detail in chapter 7 of this dissertation.

2.6 Setting: Positionings of Agile development and UX design

Reviewing the Agile/UX literature finds that discussions on the combination of Agile development and UX design draw either from practitioners' experiences in their work setting, or from principles and values of Agile development and UX design as they are found in written sources. This section discusses how the settings, i.e. places of work (e.g. organisations) or the written sources, are handled in published accounts. Discussions on the combination of Agile development and UX design is strongly influenced by practice — that is, discussions draw from the experiences of practitioners in their work settings. Published reports draw on these in various ways and the discussions they contain are based on reports on teams in organisational settings, or debates (e.g. [Nelson 2002]) and discussions (e.g. [Coatta and Gosper 2010]) that draw from experiences of practice. However, the combination of Agile development with UX design is also addressed outside of practical settings, that is, the discussion draws on principles and values of Agile development and UX design as they are found in written sources (e.g. [Blomkvist 2005]). This section shows how the literature locates the combination in and outside of practical settings and the unquestioning manner in which the literature spans both.

Accounts addressing the combination of Agile development and UX design in practical settings, are found in practitioner reports and empirical research reports. For

example, in the practitioner report by Budwig et al. [2009], the authors describe their company, "Our company has annual revenue of two billion dollars with millions of customers in many countries." The position of the *designers within the organisational structure and their relation to the developers: "The organization is split into multiple business units (BU). The authors are part of a User Experience & Design (UED) organization that supports all of these business units." This narrative is typical of practitioner reports that present the combination of Agile development and UX design as the concerted effort between two teams in an organisation. Further examples include the reports by Cho [2009], Sy [2007], and Broschinsky and Baker [2008]. Empirical research concerned with practical settings includes research by Kollmann et al. [2009] and Chamberlain et al. [2006] — both conducted with teams in organisational settings. Both reports describe the UX teams as separate from the Agile developers and via their academic analyses of qualitative data, present the experiences and challenges faced by these two groups working together.

Reports of practice are not only drawn from organisational settings. The combination of Agile development with UX design is also reported from non-organisational settings. The report by Lee and McCrickard [2007], for example, presents two case studies where the teams comprised undergraduate students and the authors acting as managers. Obendorf and Finck [2008] compare two case studies, the first conducted with undergraduate students, and the second in an organisational setting. It is important to note that combining Agile development with UX design in an organisational setting will be contending with the forces embedded within those organisations. Whereas, the non-organisational settings will see different contextual forces. Unfortunately both the reports by Lee and McCrickard [2007] and Obendorf and Finck [2008] have little to say about what differences may or may not exist, and how to take account of such differences.

Finally, the combination of Agile development with UX design is also addressed outside of practical settings in accounts that either draw on principles and values of Agile development and UX design as they are found in written sources (e.g. [Blomkvist

2005; Rannikko 2011]), or draw on a body of literature in order to say something about practice (e.g. [da Silva et al. 2011; Sohaib and Khan 2010, 2011]). Using texts such as the Agile Manifesto[1], Lindvall et al. [2002] and Boehm and Turner [2003], Blomkvist [2005] evaluate the user-centeredness of Agile development. With a conception of user-centeredness similarly constructed from (a different set of) texts. One of the conclusions by Blomkvist [2005] is that "So far, there is no predominant reason why agile processes could not be customized or adapted to UCD, or vice-versa."

Sohaib and Khan [2010] outline clashes between Agile development and "Usability Engineering" (from [Nielsen 1992]) with the aim of presenting a new approach that claims to reconcile the clashes. Their approach is not evaluated in practice, however, they conclude that "usability fits well with the agile software development." In a similar approach, da Silva et al. [2011] review the literature to find "recurring themes and patterns of the most common activities and artifacts used by teams integrating agile methods and UCD." Rannikko [2011] analyses reports from the Agile/UX literature to formulate a list of "best practices" that combine Agile development with UCD. Sohaib and Khan [2010, 2011] and Rannikko [2011] make no attempt at distinguishing between the reports or the types of accounts that are used in the analysis presented, which is typical of the current Agile/UX literature. An exception is da Silva et al. [2011] where the reports are distinguished as experimental, empirical, experience report and theoretical. However, it is not clear how these distinctions influenced the analysis presented. Reports presented from the perspective of Blomkvist [2005] is rare in the Agile/UX literature, as the combination Agile development and UX design is typically seen as a problem played out in practical settings. However, accounts may contain comparisons of Agile development and UX design based on their principles and values, before the practical issues are addressed. The disjointedness between this type of account and accounts of practice, and what this means for an understanding of practice, is addressed in more detail in chapter 7.

The lack of discussion about the role of setting in the combination of Agile devel-

---

[4] agilemanifesto.org

49

opment and UX design, is surprising considering the variation in settings that can be identified in published reports. For research results that originate outside of practical settings, there are known problems in transferring research results to practice. For example, Sjoberg et al. [2002] reported on problems relating to transferring research from artificial experimental settings to industry. Carver et al. [2003] highlighted problems with using students for software engineering, due to "the actors' different goals, expectations, and constraints." This silence has implications for what is claimed to be known about practice, specifically with respect to practical settings and how Agile development and UX design unfolds in practical settings.

2.7 Summary

This chapter presented an overview of how the combination of Agile development and UX design has been addressed and how it extends the discussion on bridging HCI and SE. Agile development supports UX design and vice versa. For example, the up-front UX design work supports building and maintaining Agile development's notion of "vision" by providing a holistic view of the software under development. In return, Agile development supports UX design with opportunities for obtaining and incorporating feedback at regular intervals. These are some of the benefits that have been identified and that can be realised if practitioners can find ways to work together. However, the problems that practitioners have to work through in order to achieve these benefits, are not trivial: Timing/scheduling issues have to be resolved, different approaches to software implementation have to be reconciled, there are power struggles and cultural differences to be overcome. No clear picture emerges for how these problems should be addressed.

This chapter has highlighted features of the discussion on the combination of Agile development and UX design that prevent clarity and solutions from converging. The first feature of the discussion involves underlying debates around requirements and design — whether requirements can be known up-front or not, and whether the user

experience can be created via emergent design. The second feature is the variation in design approaches that are being combined with Agile development. Published reports treat the types of design approaches (and Agile development approaches) as interchangeable, resulting in an assortment of underlying perspectives and assumptions that remain obscured in the discussions. The third feature is the limited attention to how setting shapes practice. This holds implications for what is claimed to be known about practice and how Agile development and UX design unfolds in practical settings.

The literature is rich in experience of practice and advice for practice. In order to proceed with an investigation into how Agile development is combined with UX design, accounts of practice from the literature require further analysis to make what has been left implicit, explicit. The next chapters present the analysis of the Agile/UX literature and the studies of teams in organisational settings. The discussion builds towards an understanding of the integration of Agile development and UX design in practice.

CHAPTER 3

Research design

3.1 Introduction

This chapter provides an overview of how the research was carried out. Two components contribute to answering the research questions: one is the investigations into practice, or field work (presented in chapters 4–6) and the other is an analysis of accounts of practice from the literature (presented in chapter 7). Throughout the course of carrying out the research these two components informed each other.

The next sections describe the methodological foundation and research design (§ 3.2), the field work component (§ 3.3) and the analysis of accounts of practice component (§ 3.4). Finally, a summary is presented (§ 3.5).

3.2 Methodological foundation and research design

In designing a course of research, it is necessary that the research approach is appropriate for answering the research question [Robson 2002, p. 80]. The overarching research question in this dissertation is *How are UX design and Agile development combined?* The approach(es) to research chosen depend on assumptions about the nature of Agile development and UX design and how they can be understood. For this dissertation, the emphasis is on understanding and informing software practice, which requires a focus on people and the settings in which they achieve their work [Lethbridge et al. 2005]. The research question encompasses notions of what problems the combination of Agile development and UX design suggests, advice on how this problem can be addressed and how Agile developers and UX designers work together.

Motivated by our research aims, this research is ethnographically-informed [Robinson et al. 2007]. Using field work and published accounts of practice as sources of data, our investigation into how Agile development is combined with UX design, took into account how the developers and designers assign meaning to what they do and how they use documents, tools and other artefacts in their work setting. The contributions presented in this dissertation are a synthesis of the results of the field work and the analysis of accounts of practice from the literature. An overview of the research design is given in figure 3.1.

The literature contains accounts of the combination of Agile development and UX design with the aim of understanding and informing practice. Such contributions include empirical research accounts and accounts of practitioners' experiences. This body of literature was recognised as an important source of data for addressing the research questions in this dissertation. Therefore, accounts were selected and analysed with respect to the nature of the challenges of combinging Agile development and UX design in practice, and the solutions from and for practice. Section 3.4 provides more detail on the analysis of accounts of practice. Studies of the reality of Agile development and UX design in practical settings (field work) complements the analysis

Field work	synthesis	Literature
Study1		Accounts of practice
Study2		
Study3		
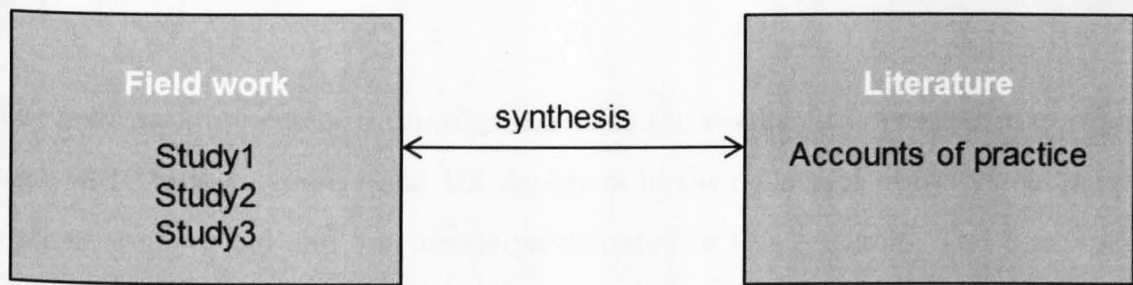
Figure 3.1: Research design.

of accounts of practice from the literature. Section 3.3 provides more detail on the field work. Throughout the study, the research design remained flexible "with the potential to change or shift emphasis and even direction as they [projects] unfold" [Barbour and Barbour 2003]. The research design was *emergent* in that participants and findings from the field work and the analysis of the accounts of practice, were allowed to direct the next steps. Focus and scope were determined as more data was generated and the analysis progressed.

What people say they do and what they actually do is accepted as not being the same [Agar 1980, p. 107] and field work can be seen as a way to access what practitioners actually do (Cockburn, reported in [Highsmith 2002, pp 80–81]). Practitioner reports are a kind of self-reporting that has been analysed as primary data in this research. It is not the view taken in this dissertation that one form of data trumps another, or that one form of data allows a more accurate account than another. The research does not compare the results of the analysis of accounts of practice from the literature and field work in order to make value judgments about them.

The role of the researcher in this inquiry is understood to be an active one: "As such, our world or our reality is always something we make, not something we discover or find [Rorty 1979, 1989]," [Smith 1992]. Mason [2002, p. 52] upholds a relationship between the researcher, the data and the social world, that suggests that data is not 'out there' in a state ready for collection. Instead, data is *generated* by the researcher participating in the social world. This is the stance taken in this dissertation, even when the term data *collection* is used.

54

3.3 Field work

The field work presented in this dissertation is the result of the researcher spending time with Agile developers and UX designers in their place of work. Three organisations were visited and four teams participated in this research. The time spent with each team varied. For Study 1, the length of time spent with the team was two weeks, six days in Study 2 and two days in Study 3. In each case, the amount of time covered a meaningful period for each team. In Study 1 and 2 the period covered was one sprint. In Study 3 the period covered was the time that the developer was sitting together with the designers in one room.

The selection of teams to study was opportunistic. This means that it was not known before setting out to find participants, who would be open to supporting the goals of this study. Individuals who were involved in organisations where Agile development was being combined with UX design were approached through research contacts. Those that agreed with the aims of this research allowed us access to spend time with their teams in their place of work. The teams studied in this dissertation are not considered to cover all the possibilities found in practice.

Details of the participants in each of the three studies are summarised in table 3.1. As shown in the table all teams studied were small to medium-sized Scrum teams developing software for the web or mobile devices, working as part of organisations varying in size. All participants were based in the United Kingdom, and embedded in established organisations that highly valued UX design. Study 1 involved a Scrum team in a large organisation, working with UX designers in the same organisation. Study 2 involved two Scrum teams and one UX designer in the same organisation. At the time of the study, the UX designer was assigned to work with only one of the teams. The team in Study 3 was not an established Scrum team. Only one individual had any experience of working as part of an Agile team. This team had chosen to trial Scrum with one of their projects and, as part of this trial, to seat the developer and designers in the same room for two days. Study 3 involved one Scrum developer and

two UX designers. Further details of the participants, such as experience with Scrum development and details about their roles, are provided in the chapters in which the studies are discussed (chapters 4–6).

The role of the researcher was one of informed observer. This falls on a continuum that can range from 'complete observer' to 'complete participant' [Junker 1952]. In the informed observer role I considered everything in the field as 'strange.' However, with a background in software engineering and UX design in practice, I was not unfamiliar with the kind of work the teams were doing. I did not know the details of how work proceeds in each setting, however, their work was recognisably software development work. The aim was to observe the natural setting of practice with minimal intrusion [Robinson et al. 2007]. The researcher engaged with the participants, asking questions about the work and conducting one-on-one interviews as appropriate. The participants were made aware of the research objectives and that their data was to be written up in a PhD thesis and published at appropriate conferences/journals.

Table 3.1: Participant overview

| | Study 1 | Study 2 | | Study 3 |
	Team1	Team2A	Team2B	Team3
project type	web	mobile	mobile	web
team size	16	6	7	4
Agile method	Scrum	Scrum	Scrum	Scrum
UX role	yes	no	yes	yes
number of people*	> 1,000	< 50		± 50

* The numbers are an estimate and are intended for comparative purposes only (taken from publicly available information about the organisations).

3.3.1 Main events

How the field work fits into the larger research design has been shown in figure 3.1. This section explains how the individual studies were carried out, in terms of the order in which they occurred, the main events for each and how each was informed by the findings from the former. The main events for each study were: (1) data collection, (2) preliminary analysis, and (3) feedback session. The chronological order differed from how they are presented in this dissertation. The data collection, preliminary analysis and feedback session for Study 1 was completed before the next round of data collection. Chronologically, Study 3's data collection occurred next, followed by the data collection for Study 2, etc. The order in which the studies are presented in this dissertation, i.e., Study 1, Study 2, then Study 3, emerged from the ongoing analysis and synthesis, which suggested that how the developers and designers in Study 1 achieved the integration of Agile development with UX design, differed in important ways from how the developers and designers in Study 2 and Study 3 achieved integration.

1. **Data collection.** This was the stage at which the observations and interviews were conducted. The observations were the main source of data collection. All interviews were conducted with the aim of learning about the work being observed. Data was collected at the work premises of the Agile team under study. During the course of the observation period, the observer asked the participants questions for clarification. Due to time constraints the issues that needed more in-depth questioning was scheduled as part of a separate interview with an individual participant. The exception was in the case of Study 1 — the interview with the UX designer was conducted on the day of the feedback meeting. This was due to matters of access.

2. **Preliminary analysis.** The preliminary analysis comprised the early stages of the thematic analysis and generated the initial themes, which were refined throughout the process of analysis. The preliminary analysis was carried out to

the point where the initial themes that emerged from the data could be shown to the participants and discussed with them in a meaningful way. The results of the preliminary analysis were discussed in the feedback sessions in each study.

3. **Feedback session.** The feedback session served two aims. First, it was a strategy to convince practitioners to take part in the study. For providing the researcher access to their teams, the researcher could in return reflect back what had been observed and inform participants how their approach compares with previous research and the literature. Second, the feedback sessions served as opportunities for the researcher to check with the participants that the meanings and purposes of the work observed had been understood. Participants' reactions to the feedback sessions were very positive and they used it as an opportunity to reflect on their situation. The results of engaging with the participants in the feedback sessions informed the analysis of the data.

While each study had commonalities in the method, there were differences with respect to when and how the data was collected and analysed. Each study contains a description of those details in the chapter in which they are presented in this dissertation.

3.3.2   Thematic analysis

Analysis of the data from the field work was not a distinct stage and overlapped with data collection and events in the other studies. As the research effort progressed, the data analysis was informing and was being informed by the data collection and feedback events from the other studies. Successive rounds of writing, discussing, reflecting and re-writing, shaped the structuring of patterns emerging from the data into themes and shaped the relationships between themes. A theme "captures something important about the data in relation to the research question" [Braun and Clarke 2006]. For example, in Study 3 the theme *(Re)creating progress* emerged from identifying features of the interactions between the developer and designers in Team3. Across

the whole data set for Study 3, we searched for events relating to when they talked to each other and what they talked about. These events indicated that the team communicated when they encountered uncertainties and were relying on each other to resolve uncertainties and move forward in their work. In the field notes, examining exchanges such as:

> [3.FN.1.24][1] *[information architect]*: "How many projects did they say they..."
> *[developer]*: "A few every week."
>
> [3.FN.2.3] *[developer]*: "You can't always remember everything — we need to work together especially to do QA."
>
> [3.FN.1.11,12] *[information architect]* to *[developer]*: "Would that work?" *[developer]*: "You'd be limited ...that would be more development time."

allowed the researcher to characterise the content of the exchanges and to describe them as presented in sections 6.4.2 and 6.4.3, and figure 6.3. The ways in which the developer and designers practiced Scrum development, engaged with each other, made decisions, made progress and shaped the dependencies between themselves could be described. These (sub)themes comprise the Findings section of Study 3 and are required to build up the theme *(Re)creating progress* in the Dicussion section of the chapter in which Study 3 is presented (chapter 6). Similarly, the (sub)themes in the Findings section were important for building the theme *Expectations about acceptable behaviour (work group culture)* (§ 6.5.2). The presentation of themes follows the same structure across all three study chapters. Each theme reported in this dissertation contributes to answering the research question, *How are UX design and Agile development combined?* in terms that explicate how the developers and designers worked together and why they worked together in the ways that they did.

---

[1] The reference at the beginning of each quote is an identifier that allows tracing the quote back to the original artefact it has been taken from. The format is [study number.artefact name.date or day number.page number(.team name as appropriate for Study 2)]. The abbreviations for artefact names used in this dissertation are FN for field notes, IN for interview notes or transcripts, and FB for feedback notes or transcripts.

The data for this research was collected in order to be treated qualitatively. Assigning numerical values to the data was not an intended aim. Quantitative results from qualitative data are not unheard of in thematic analysis, however, transforming qualitative data into quantitative results is not unproblematic [Sandelowski et al. 2009], and the value of doing so is still unclear [Braun and Clarke 2006]. The nature of what we decided would address our research questions was inherently social. Consequently, the meanings that people assign to the work they do through their language and actions were the focus of the field studies. The significance of the results in this analysis are also not based on quantitative aspects — how many times an instance occured did not make it more or less important for answering the research question. Instead, themes were judged to be significant based on their *explanatory power.* That is, significance was judged according to the themes' ability to account for how and why the developers and designers worked together in the ways that they did, with the relevance of the themes determined in the early stages of the analysis via member checks (see Lincoln and Guba [1985, p. 314] for a discussion on member checks).

The themes presented in this dissertation were refined in consultation with my supervisors. All findings are sustained by the detailed data and have been evaluated in some form by the participants, the researcher and practitioner communities. The particulars of the data collected in each setting is discussed in the individual study chapters (chapters 4–6) and summarised in table 3.2. Moving from the raw field data to the presented themes in this dissertation followed a process of analysis that can be described on three levels. The first level generated the initial themes from the field data. The second level refined the initial themes and involved member checking. The third level developed the refined themes further into the themes presented in this dissertation. During the thematic analysis, various supporting artefacts were generated. Table 3.2 summarises the supporting artefacts for each study and the stage of the analysis at which the artefacts were generated and used. The details of each level of the analysis are as follows:

reflecting

Field data ⟵⟶ **Initial themes**

Organising
Discovering, sorting and clustering
related instances
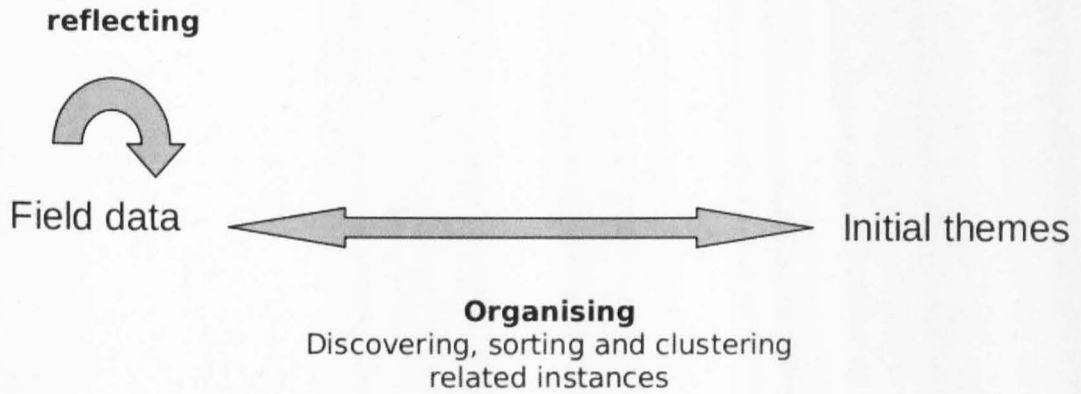
Figure 3.2: Level one of the thematic analysis of the field data: Generating initial themes.

The first level of the thematic analysis is represented in figure 3.2. Analysis at this level involved reflecting on the primary data as it was being generated in the field. That is, as the researcher observed the participants, the researcher was actively reflecting on the field notes (see figure 3.3), or notes of the interviews already recorded, to decide what to pay attention to, who to talk to, and what further clarifying questions to follow up with. Once the observations had been completed, all the data collected was organised into initial themes by discovering, sorting and clustering related instances. The initial themes were collected in an electronic document. An extract of the document with the initial themes for Study 2 is shown in figure 3.4. As can be seen in the extract the *QA* theme emerged as an initial theme, which groups together all the events in the raw data relating to what the researcher noted the individuals in the QA role as doing during the day, who they were speaking to and what their conversations consisted of. This theme was refined at levels two and three of the analysis.

The second level of the thematic analysis is represented in figure 3.5. Analysis at this level involved further reflection on the initial themes and scrutinising the field data for confirming and disconfirming instances. "Disconfirming instances" are data that contradict a theme. When a disconfirming instance was found, we investigated the events in our data surrounding this instance, in order to understand under what circumstances the contradiction occurred. In this way, the themes were refined iter-

Figure 3.3: Two artefacts from the field studies. On the left is a page from the researcher's note book showing some of the field notes taken on that day. On the right, artefacts created by the participants in Study 2 during a retrospective meeting.

atively — either confirming or disconfirming through constant comparison between instances. Continuous organising and re-organising of the themes also contributed to their refinement. At this level, the themes were presented at the feedback sessions with the teams we studied. Figure 3.6 is an extract from the feedback session held with participants in Study 2. As can be seen in the extract, the researcher confirms with the scrum master the prominent role of QA in both Team2A and Team2B (since *QA* emerged as one of the initial themes). Themes were also presented to practitioner and researcher audiences in publications (see section 1.6 for publication details). The feedback from these activities then informed further refinement and validation of the themes.

The third level of the thematic analysis is represented in figure 3.7. Analysis at this level involved building up a narrative around the themes by referencing current literature. The aim was to gain new perspectives on the themes generated by this research and to establish their significance in the context of the current Agile development and UX design literature. The results from this are written up in the discussion sections of the individual study chapters (§ 4.5, § 5.5, § 6.5) and continued in the discussion chapter of this dissertation (see chapter 8). For example, the *QA* theme in Study 2 is related to the Agile literature on roles in Agile teams as discussed by Schwaber [2004] and Beck [1999], then also compared with findings from empirical research by Martin [2009] and Hoda [2011]. The narrative was developed in successive drafts of the dissertation chapters and drafts of the published papers listed in table 3.2. On-going organisation of the themes and continuous scrutinising of the data refined the themes such that the themes presented in this dissertation, are validated and grounded in the data.

The analysis identified patterns, or themes, that related to (1) the activities of the UX designers and developers (2) how and when their activities were performed (3) how and when they communicated with each other (4) what they talked about and (5) any further aspects of practice that emerged as relevant to integrating UX design with Agile development work. The analysis was data-driven. That is, the analysis

did not involve applying a pre-defined coding scheme or testing hypotheses.

3.3.3 Summary of field and follow-up data

In summary, Study 1 generated 61 pages of A4 field notes covering approximately 67.5 hours of observations, three hours of interviews, and a one hour feedback session. Study 2 generated 20 pages of A4 field notes covering approximately 42 hours of observations, one hour of interviews, 537 lines of Skype chats, and a one hour feedback session. Study 3 generated 48 pages of A5 field notes covering approximately 15 hours of observations, and a one hour feedback session.

3.3.4 Ethical considerations

Ethics approval was obtained from The Open University's ethics committee for carrying out this research with human participants. Appendix B includes copies of the approved application form, as well as the information sheet and consent form provided to the participants.

3.4 Analysis of accounts of practice

The *analysis* of accounts of practice presented in this dissertation is a primary research activity standing apart from the literature *review*. Whereas the literature review in this dissertation shows that there is a need for carrying out the research, the analysis of the accounts of practice sets out to address the research questions and make explicit what published accounts leave implicit. The question guiding the analysis was:

> *What are the existing perspectives on combining Agile development and UX design in accounts of practice in the literature?*

Which was further broken down into:

- How is the issue of combination conceptualised and how is it addressed?

Artefacts	Study 1	Study 2	Study 3	Thematic analysis
Field data	Field notes, photographs, interview notes	Field notes, retrospective meeting posters, interview notes, Team2B's Skype chats, photographs	Field notes, photographs	Generated at level one
Follow-up data	Notes of feedback session, interview transcript with UX designer	Transcription of feedback session	Transcription of feedback session	Generated at level two
As a result of analysis, researcher/practitioner feedback	Documents with various stages of thematic analysis, supervision minutes, drafts of SPE paper, SPE paper reviews, drafts of XP research paper, XP research paper reviews	Documents with various stages of thematic analysis, supervision minutes, various drafts of the thesis chapter, drafts of XP research paper, XP research paper reviews	Documents with various stages of thematic analysis, supervision minutes, various drafts of the thesis chapter, drafts of XP experience report, XP experience report reviews	Generated at level one, two and three

Table 3.2: Artefacts generated during the field work and thematic analysis.

- What is the experience of using Agile development and UX design together in practice?

The aim was to gain a sense of what accounts in the literature were presenting as "problems" and "solutions" concerning the combination of Agile development with UX design and the circumstances in which these problems and solutions are presented. We were interested in how the authors assigned meaning to Agile development and UX design practices as they were reported in the accounts. The analysis of the accounts of practice started with a comprehensive search for relevant accounts on a narrow topic — on Agile development and UX design practice — selected according to specific criteria, which were then thematically analysed. The analysis borrows from the methods of systematic literature reviews (as described by Kitchenham [2004]), but differs in important ways because our interest was in analysing the papers as accounts of practice rather than aggregating evidence to answer a specific question.

Systematic literature reviews proceed according to a review protocol (a detailed plan for conducting the review) a defined search strategy, inclusion/exclusion criteria and strategies for obtaining the required information from the selected studies [Kitchenham 2004]. Each step is rigorously documented. The analysis of accounts of practice presented in this dissertation was conducted in a similar way to that of a systematic literature review — in that searching and analysing the literature was strategic and documented. The method used in our analysis allowed "order" to emerge from an inductive analysis of the accounts. That is, the analysis was not hypothesis-driven nor was it imposing pre-defined categories on the data that did not emerge from the data. The research questions were addressed by developing the themes that emerged from the data. Whereas systematic literature reviews typically include literature that adheres to certain benchmarks (e.g., rigour, credibility and usefulness in the systematic literature review by Dybå and Dingsøyr [2008]), our analysis focused on the descriptive qualities of the accounts, their similarities and differences. It is therefore significant that the texts included were analysed as *accounts*, rather than *studies*, and that the criteria they needed to satisfy were that the accounts contain

enough detail about Agile development and UX design practice and enough details on how practice proceeded/could proceed.

The primary source of data for the analysis was the published, peer-reviewed accounts by the professional and academic communities contributing to knowledge on combining Agile development and UX design. Electronic sources, such as ACM Digital Library (`http://portal.acm.org/`) and IEEE eXplore (`http://ieeexplore.ieee.org/`), were queried and the texts were selected using a rigorous protocol. The protocol selected texts with the combination of Agile development and UX design in practice as the main topic of discussion. In summary, 29 practitioner reports and 23 empirical studies, published between 2002 and 2010, were included in the analysis.

Kitchenham [2004] states that "As yet, there is no definitive method for accumulating evidence from studies of different types." The approach taken to analysing the literature in this dissertation was to thematically analyse practitioner reports and empirical studies separately and in detail first, and then to compare the findings. Further details on how the accounts were selected and analysed are given in chapter 7.

3.5 Summary

This chapter has introduced the research design and the strategy for synthesising the various sources of data. Findings from the two ethnographically-informed, independent components — the field work and the analysis of accounts of practice from the literature — contribute to answering the research questions. The field work consists of three studies of teams in organisational settings. The analysis of accounts from the literature involved analysing published accounts of practice. Both components involved thematic analyses of the data. The findings from the analysis of the accounts of practice are also compared to the findings from the field work in chapter 8. While this chapter has given an overarching view of how the research was conducted, further details particular to each research activity are given in the individual chapters (chapters 4-7).

> *Team2AProject*
>
> [2.FN.3/06.1.Team2A] at retrospective: Group 3 2nd level team: need help with subversion. iPhone stuff. Want more training and time to read books.
> [2.FN.17/06.17.Team2A] during Team2AProject demo: *[developer]*: Lots of problems with build sequence and producing interaction with sound.
> [2.FN.17/06.20.Team2A] at retrospective: Talking about problems they spotted vs. Things they didn't know they needed, e.g., some ending videos for Team2AProject. Wireframes were delivered but there were many changes and they were never redelivered.
> :
>
> *Team2BProject*
>
> [2.FN.17/06.17.Team2B] during Team2BProject demo: *[developer]* reporting with actual handset. Everyone crowds around handset that *[QA]* has. Shows sign-in → has option of doing the upgrade. If yes to 'Allow application to access the web?' hit yes, otherwise security failure. Prompt: do you want to replace x.y.z with a.b.c? Hit yes. *[developer]*: basically always first checks for an upgrade. Saves username and password. Showed the optional version. Now showing mandatory version.
> [2.FN.17/06.18] at the skiff after gold card discussions: *[Scrum master]*: *[client]* want to plug social networking into Team2BProject software. Want to get both teams on to Team2BProject. Waiting until September *(when Team2BProject might be finished)* before taking on full-blown new project.
> :
>
> *QA*
>
> [2.FN.3/06.3.Team2B] during Sprint planning meeting: Ensuring QA has something to do at the beginning of the sprint.
> [2.FN.5/06.6.Team2B] during standup: *[UX designer]* mentions another test case behaviour: don't know yet. *[product owner]*: Still waiting on response for some questions from *[client]*. *[QA]*: kind of hard to do QA if we don't have the answers. *[QA]*: wary of starting stories without behaviours which means we might have to change things and that wastes time.
> [2.FN.15/06.25.Team2A] *[QA]*: can I have a look at that quiz now? *[developer]*: Not sure if I got the assets in there ...
> :

Figure 3.4: Extract from document containing the raw field data sorted into initial themes for Study 2.

Evaluation by Research/practitioner community

reflecting

Publications

Initial themes ⟷ Refined themes

Organising
Sorting and clustering
related initial themes

Scrutinising the data

Feedback session

Member checking
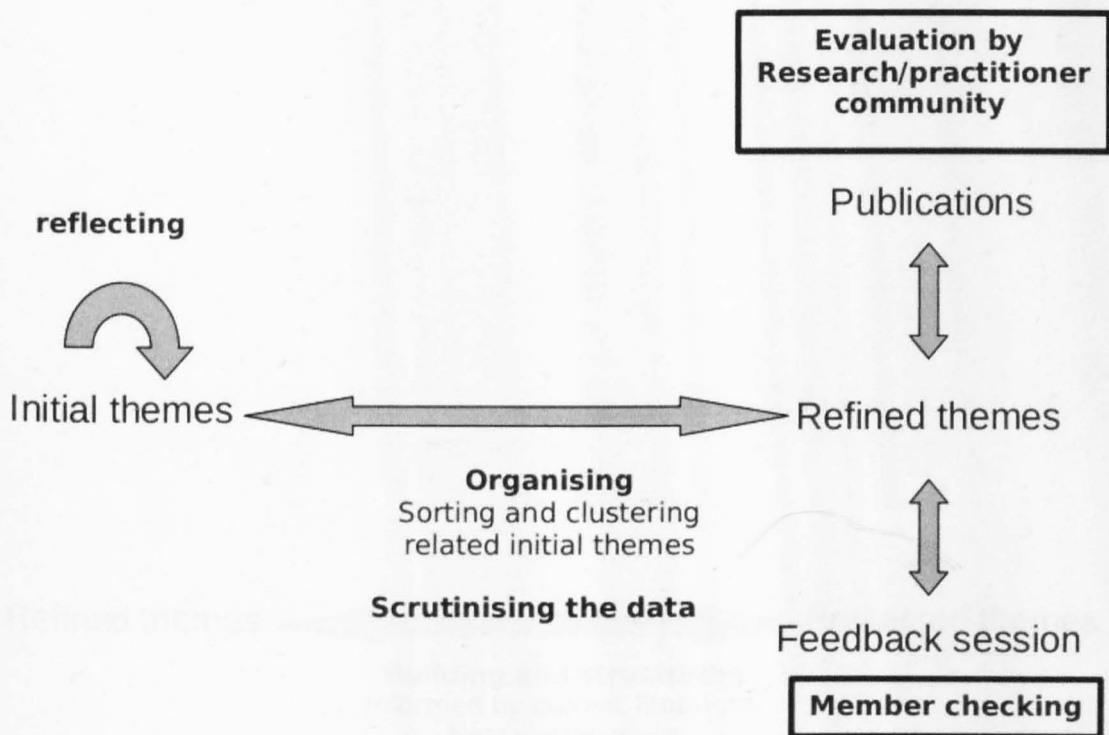
Figure 3.5: Level two of the thematic analysis of the field data: Refining the initial themes.

> **researcher**: So to me it appeared that QA were testing UX related issues
> as well. I heard feedback from *[QA]* saying "Is this the right colour?"
> **scrum master**: Certainly I'd see QA as testing not the code but the whole
> product. And one of the things that we've noticed — it hasn't been delib-
> erate — but one of the things that we've noticed is that because QA are
> using phones all the time, they've got a wide variety of experience with dif-
> ferent devices, they're often the people noticing UX problems and one of the
> reasons why we try to get them more involved with the UX side of things.
> :
>
> **researcher**: I think QA, ... they're having conversations with the Product
> Owner and UX, they all have input into the design of the product.
> **scrum master**: Yeah and I'd like to think that everyone, developers, de-
> signers, me and *[business development manager]* also get involved with the
> design process ... on the one hand we've always found that quite good. If
> you involve someone in the process then when the time comes to build it,
> they feel "Yes, this is something I've done." If you don't give them any
> involvement then the minute they find a problem, A they've lost a lot of
> confidence in the design team and B they're sitting there doing something
> which they don't believe in themselves. Which never produces the best re-
> sults so they do a better job when they're actually involved in the design
> themselves.

Figure 3.6: Extract from transcript of feedback session for Study 2.

Refined themes ⟵――――――――――⟶ Presented themes

Building and structuring
Informed by current literature:
New perspectives
Significance of the themes
Context of Agile development and UX design

Organising
Sorting and clustering
related themes

Scrutinising the data
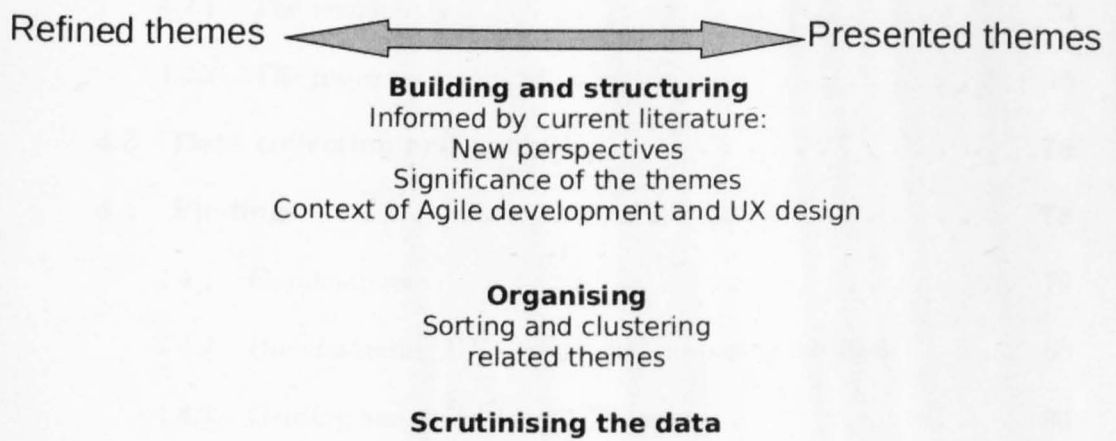
Figure 3.7: Level three of the thematic analysis of the field data: Presenting the refined themes.

CHAPTER 4

Study 1

4.1 Introduction to Team1

This chapter presents an ethnographically-informed study of an established Scrum team in a large organisation, working with UX designers[1] in the same organisation. Although the designers and developers were not considered part of the same team by the organisation, we refer to them as Team1 in this discussion, based on their dependencies in carrying out their work. The observations covered a two-week sprint.

The next section (§ 4.2) presents background information on the people, the project they were working on and their organisational setting. Next, data collection and analysis details are discussed (§ 4.3). Next, the themes that emerged from the analysis are discussed (§ 4.4): the activities that developers performed in order to turn the visual designs and wireframes into working software (§§ 4.4.1–4.4.3), and how these activities were the innovative way the developers and designers were working amid a culture of separation in the organisation (§ 4.4.4). These findings are discussed (§ 4.5): the situated nature of Agile development and UX design work (§ 4.5.1), Agile development and UX design as cooperative work (§ 4.5.2) and how cooperation was managed through articulation work (§ 4.5.3). Finally, the key findings of Study 1 are summarised (§ 4.6).

4.2 Background

Team1 were part of a large media organisation in the UK. The team of developers was one of several Scrum teams in the organisation responsible for the organisation's digital media services, including creating and maintaining the organisation's public-facing website. At the time we arrived on the site, the team had been working on this project for one year and the project had been through several iterations (both of the

---

[1]Section 4.2.1 explains what is meant when using the term *UX designers*.

code and the UX design). The implementation work that required little or no UX design had been largely completed and the Agile developers were waiting for the UX designers to hand over the final versions of the wireframes[2] and visual designs[3] for implementation. There were two two-week Sprints remaining before the first release to end users, when this study began.

The developers and designers were seated in open-plan office space on separate floors located in the same building. Figure 4.1 shows the developers' work space and the wall and stories. The UX designers were located on the floor above the Agile developers. The researchers were seated with the Agile developers during observations (unless attending meetings elsewhere), since access for the observations had been negotiated with the project manager of the Scrum team. However, the researcher was given access to planning meetings arranged between the developers and designers.

The Scrum team's sprint began with the sprint planning meeting and ended with a retrospective meeting. At the start of each day, the developers attended a stand-up meeting to discuss the work for the day. The UX designers did not attend sprint planning, retrospectives or stand-ups. There was a daily meeting scheduled between representatives from the UX designers and developers (called a UX scrum), but during the time of the observations, it became evident that these meetings were not addressing the developers' day-to-day questions regarding the UX design. The rest of this chapter shows how the work of the developers depended on frequent interactions with the UX designers.

From the start of the observation period it was notable how the work of the developers included finding UX designers for decisions and clarifications. The developers sent emails but also walked around to find a UX designer. In the developers' attempts to make contact with the UX designers there were times when the designers were not

---

[2]By *wireframe* we mean the popular design tool, which conveys content and layout information, as described by Garrett [2002]. In this study the wireframes were paper-based representations of the website under construction.

[3]In this case, the *visual design* was a paper-based representation of the website under construction, and conveyed the content and layout information along with graphical aspects of the design.

Figure 4.1: The image on the left shows Team1's work space and the image on the right shows the wall and stories.

available to respond. Throughout the field notes there are examples of this. On the first day:

> [1.FN.1.4] 2pm: There's some problem with the visual design from UX [hadn't arrived yet]. Project manager's been trying to get in touch with the interaction designer, however can not get in contact — mobile switched off.

The tensions between the UX designers and developers were explained to the researcher from the first day on site:

> [1.FN.1.1] Some time after 11am, in a conversation between the researcher and the Project manager: There's a lot of tensions between UX designers and developers ... UX people want to own the requirements ... they take very long to deliver their designs, while the developers are just waiting.

4.2.1 The people

Our study involved 14 Agile developers who were part of the *Internet division* in the organisation. The Agile developers had been working together and developing software using the Scrum methodology for approximately one year. They were collectively known as the "developers" and referred to themselves as such, however,

Role	Individuals
project manager/Scrum master	1
product manager/product owner	1
client-side developer	2
software engineer	8
tester	2
interaction designer	1
graphic designer	1

Table 4.1: The roles and number of individuals for each role.

individuals in the team had their own role according to their area of expertise. The roles were project manager/Scrum master, product manager/product owner, client-side developer, software engineer and tester. Among the developers, experience with Scrum development varied   as a development team they had been doing Scrum development for approximately a year, whereas the product owner had approximately three years of Scrum experience.

Separate to the Internet division, a substantial *User Experience division* consisted of information architects, interaction designers and graphic designers. There was one interaction designer and one graphic designer associated with the same project the developers in Team1 were working on. The interaction designer was responsible for creating wireframes, while the graphic designer was responsible for creating the visual design. For ease of discussion, we refer to the interaction designer and graphic designer as the *UX designers*. The developers were responsible for implementing features of the website as working software and were reliant on the UX designers to hand over the appropriate designs at the appropriate times, such that the features could become part of the working software released to the client. The roles and the number of individuals in each role are shown in table 4.1.

4.2.2 The project

The UX designers were not dedicated to a single project, but were spread across various projects. At the time of the observations, the Agile developers were working on

a project implementing sign-in and registration features for a part of the organisation's public website, as well as migrating existing user registration details to a more secure server. The UX design for these features was expected to become the standard for registration and sign-in across the entire website. During our time with the developers, they were concentrating on implementing a web form that allowed the user to register with the site. The form contained fields for entering the user's personal details, including name and date of birth.

4.3   Data collection and analysis

For the duration of one sprint (ten work days) the researcher observed the day-to-day activities of Scrum development. Figure 4.2 shows the timing of the observations with the developers' sprint. The researcher attended team meetings and was co-located with the team in their work area during their normal hours of work. The meetings attended were the sprint planning meeting at the beginning of the sprint, the daily stand-ups, the retrospective at the end of the sprint, as well as meetings called between the developers and the UX designers during the sprint. The aim was to capture salient episodes as they happened and this was done via contemporaneous field notes hand-written in a notebook. Notes were made of the following:

- naturally-occurring conversations between participants,

- conversations between the researcher and participants — including instances where explanations were spontaneously offered by the team members, as well as where the researcher prompted for explanations,

- descriptions of individuals' movements — for example, moving between desks, or going up and down stairs,

- sketches of the seating arrangements,

- descriptions of tasks, and

- notes about documents, emails and tools that were shown to the researcher. Due to confidentiality agreements, the researcher was not able to remove any artefacts from the work premises.

Data collection oriented towards instances where the developers were either in direct communication with the UX designers, or were focusing their attention on UX design issues. An illustration of the data that was collected in the field notes for this study is presented in appendix C. During the observation period it became apparent that the key team members, in terms of who carried out the work of integrating UX design and Agile development, were the developer roles interfacing most frequently with the UX designers. Interviews were therefore conducted with a client-side developer and the product manager/product owner. These semi-structured, one-to-one interviews were conducted in a separate meeting room, at an agreed time with questions that were based on the observations. The interviews gained more background on the interviewees and their accounts of how previous events and forces within the organisation were shaping what was being observed. For example, these interviews elicited accounts that supported the observed separation between the developers and UX designers   the interviewees talked about the "schism" and the "us-versus-them" that they felt existed between developers and UX designers in that organisation. These interviews were recorded via hand-written notes.

As the observations captured practice mostly from the developer perspective, the researcher returned to the work premises after the observation period was over, to conduct an interview with one of the UX designers (the interaction designer). The interview coincided with the feedback session (§ 3.3.1). The aim of the 45 minute long semi-structured interview was to gain more insight into the UX perspective on issues that had emerged during the preliminary data analysis. This interview was voice recorded and transcribed.

The observation period was spent with the Agile development team and the analysis reflects this. The findings of the data analysis have been made available to the

Day1 Day2 Day3 Day4 Day5 Day6 Day7 Day8 Day9 Day10

| Start of sprint | End of sprint |

Team1's sprint
observed
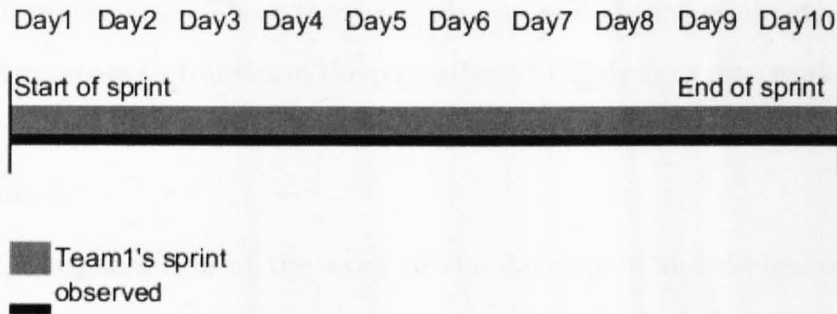
Figure 4.2: An image showing the timing of the observations with the sprint.

participants in this study, who were invited to comment. The feedback from both the UX design and Agile development perspectives have subsequently been incorporated into the analysis presented here.

4.4 Findings

Although the detailed analysis of accounts of practice had not yet been completed when the observations for this study were conducted, preliminary analysis of the literature had shown that discussions concerned with combining User Experience (UX) design and Agile development have either focused on integrating them as two separate processes, or on incorporating techniques from UX design into an Agile context. An academic view on the nature of practice in this area was limited. The many and varied settings in which Agile developers and UX designers work together, and how those settings shaped their everyday work, were largely unexplored.

The perspective taken in this study was strongly influenced by a knowledge of the literature, that is, it is a perspective on the work of the UX designers and Agile developers, in terms of *activities* and how they *coordinated* their activities on a day-to-day basis. By *coordination* we mean "managing dependencies between activities" [Malone and Crowston 1994]. The work of interest during our observations was all the activities associated with transforming the paper-based designs produced by the UX designers into working software. The themes that emerged from the data are

presented in this section. These themes centre around the types of activities that allowed the developers to transform the paper-based UX designs into working software. A cross-cutting theme was related to the organisational setting in which the activities were performed.

While the organisation of the work of the developers and designers was agreed within their respective divisions, e.g. the developers followed the rules of Scrum development, there was no formalised, agreed model according to which the developers and designers coordinated their work with one another. These activities were essentially uncovering the need for further input from the UX designers. Next, these activities are placed within the wider organisational setting.

4.4.1 Gap analysis

The phrase "gap analysis" was used by the developers to refer to an activity which involved inspecting the designs handed over by the UX designers, comparing it to the software already implemented and identifying mismatches between them. Mismatches also appeared when the designs were translated into working software implementing a paper-based design often uncovered areas of the design that could not be directly implemented as working software because it required more detailed design, or a redesign. In one instance from our observations of a gap analysis, the developers realised that the UX designers had included a field for entering a date of birth as a list box, which was a change from the previous version of the design in which the field for entering a date of birth had been a text field. This design change impacted on the format of the date as it was currently stored in the database of registered users. Recognising this conflict, the developers agreed that before implementing the design they would have to approach the UX designer responsible for the design to ask for a clarification of the decision regarding the change. Another example of an outcome from the gap analysis was the developers realising the description of the dynamic aspects of the design provided in the UX design documents were not suffi-

cient for implementation. Again, the developers' realisation prompted further action to obtain a more detailed explanation from the UX designers.

4.4.2   Understanding UX designs and preparing feedback

The developers were observed investing time in becoming familiar with the UX designs that were sent to them, building up an understanding of what the designs meant in terms of working software. An example of this took place on a day when the developers received the UX designs emailed to them in the form of a pdf document. When the developers noticed the design documents in their mailboxes the group of developers crowded around one person's monitor and discussed among themselves what the design meant. They asked each other questions such as: Was this the final design? Were they going to be able to provide feedback on this design and have that feedback incorporated? What does that comment mean? Why does it look like that? How are we going to do that? After a lengthy discussion the developers had arrived at some understanding of what the designs required in terms of implementation work. They were now in a position to provide feedback to the UX designers in the form of more questions and clarifications, such that once they had the UX designers' response to their feedback they could proceed with the implementation. The developers' feedback was then either written up in an email, or verbalised during an *ad hoc* "feedback meeting" between the developers and the UX designers.

4.4.3   Carding and prioritising UX design

The developers could not directly implement the UX designs that were sent to them in the form of pdf documents. Part of the UX designs had already been implemented as a result of previous Sprints and the work in progress in the current sprint. The developers, therefore, performed an activity that identified the remaining implementation work according to the latest version of the UX designs. This activity was similar to the "gap analysis" explained above, but whereas the gap analysis aimed at finding

mismatches between the current working software and the UX designs, this activity aimed to identify the parts of the UX design that had not yet been implemented as working software. Once the UX designs had been delivered to the developers and the developers felt they had enough information regarding the design to implement it as working software, the implementation work had to be added to the already existing Scrum backlog; that is, the implementation work had to be broken up into story cards and prioritised along with the existing stories in the Scrum backlog. One developer was responsible for systematically going through the design and breaking its implementation tasks up into story cards. The priority of each card was then discussed by the team and inserted into the Scrum backlog according to its priority. This activity was performed once the designs were well understood in terms of the implementation details, however, the UX designers were approached for design direction when unanticipated implementation issues arose. The developers found that implementing a design led to additional tasks for which the input of the designers were required to progress the implementation work. Story cards were created as these additional tasks emerged, prioritised by the team and added to the Scrum backlog.

4.4.4   Working together in a culture of separation

Up to this point the focus has been on the activities the developers were observed to carry out as they proceeded to transform the UX designs into working software and integrate that with the existing code base. The significance of the setting in which the above activities were taking place, becomes evident in the timing of the events that occurred over the course of the sprint we observed.

In the particular sprint being observed, the UX designers had promised to deliver the UX designs before the developers' sprint planning meeting. Instead, versions of the UX designs were sporadically sent to the developers during the sprint up until the last day of that sprint. The consequence was that the developers did not have the UX designs at their sprint planning meeting and could not prioritise and plan

the UX design-related implementation work for that sprint. During the sprint we observed the UX designers promising to deliver designs by certain dates that were repeatedly missed. Much of the gap analysis, understanding the wireframes and preparing feedback, and carding and prioritising the design had to be performed by the developers in the last few hours of the sprint, in order to have working software by the end of the sprint. The lack of planning upset the work that had already been planned for and the developers had to work overtime to complete the work by the end of the sprint. When one of the UX designers was questioned about whether they were aware of the implications of their timing, they responded that they had not realised that sending the UX designs when they had done was causing any problems:

> [1.IN.20/02/09.5] *[UX designer]*: "That was never explicitly said to me. So maybe they didn't even realise it themselves I guess, or they didn't think that the solution was to tell me, or whatever ..."

We observed that developers and designers made decisions only within their disciplines. Only as a last resort, in the absence of UX design decisions for some features they were working on, would the developers improvise UX design solutions or hold off implementation altogether. The developers were not comfortable with improvising their own UX designs, as in the past this had led to wasted effort when the UX designers suggested changes. The developers performed explicit actions to communicate with the designers about the mismatches and unimplementable aspects of the UX design, i.e., the developers

- requested a feedback meeting with the UX designers,

- physically moved around the building to find UX designers for decision-making,

- sent emails with feedback to the UX designers,

- met face-to-face with UX designers.

Once, in their efforts to obtain UX design decisions, the developers were told that the UX designers were unavailable due to the UX design division moving desks. Another

time the UX designer was unavailable because they were working from home. In one of the successful attempts to communicate with the designers, the UX designers expressed surprise at hearing that their designs had generated questions and that they may have to redesign some aspects. When the developers asked to have a feedback meeting with the UX designers, the UX designer responded in a tone that conveyed mild annoyance,

> [1.FN.2.6] "The visual designs are pretty much parallel to the wireframes . . . what kind of feedback do you want to give?"

The activities of the Agile developers and UX designers were shaped and sustained by the values and assumptions promoted by the organisation in which the Agile developers and UX designers were embedded. The values and assumptions referred to here are those held by management about how best to combine the skills of Agile developers and UX designers to create quality software. One of the participants in our study informed us that the decision to separate UX designers and developers was put into effect by management in the UX division. This was based on their view that UX designers work best when they are separated from the issues of software construction, which hamper their creativity. At the time we visited this organisation, the manager in charge of the UX division was seen as more powerful than the manager in charge of the developers' division. Consequently, the UX designer and developer groups were kept apart; in terms of seating arrangements, as well as decision-making. They were seated on separate floors of the same building and decision-making remained firmly within their own domain — UX designers made decisions regarding the UX design and developers made decisions regarding coding issues. The reluctance of the developers to improvise solutions can be explained by the power the UX designers had to insist that the developers redo parts of the already implemented software. To avoid redoing work, the developers opted instead to invest time and effort in attempting to have discussions with the UX designers as they encountered UX design-related issues in their implementation work.

4.5 Discussion

A picture emerges concerning work arrangements, dependencies and mechanisms that make the integration and coordination of work possible. This account is not explained by rationalised accounts of Scrum nor UX design, rather, it shows a situated form of organising work between two groups who were considered separate by themselves, as well as the organisation in which they were embedded.

4.5.1 The situated nature of UX design and Agile development

The previous section related how the developers and designers organised their work in order to turn the paper-based UX designs into working software during the course of a sprint. What our discussion of the integration activities[1] has shown, is that it was the developers who were concerned with working through the details of implementing the UX designs as working software. They were questioning and experimenting, they were matching the designs with their knowledge of the existing software, which enabled them to uncover the mismatches between the existing code and the designs, as well as the unimplementable aspects of the UX designs. In doing this, the developers reached points in their work where it was necessary to have discussions with the designers (to gain clarifications or redesigns), in order to continue with the implementation work. So in their attempts to transform the UX designs into working software, the developers also uncovered the points where involvement from the designers was required.

Despite having no model or plan for how the Agile development and UX divisions ought to coordinate their work, i.e., no organisational-level commitment to guide their coordination, the developers and designers were working to meet their respective divisional-level commitments. On the divisional-level the developers were committed to maintaining the rhythms of Scrum development, i.e., cycling through planning, implementation and evaluation activities. Similarly, the UX designers were

---

[4]From here we will refer to the gap analysis, understanding the UX designs and preparing feedback and the carding and prioritising UX design activities as 'integration activities'. This should be understood as shorthand for 'integration and coordination activities'.

working according to their own rhythms and were committed to maintaining those. Their separation at the divisional-level was supported and sustained by the values and assumptions within the organisation — values and assumptions about how the developers and designers will best produce quality software. Combining their work in the culture of separation required that individuals from the divisions step outside their divisional boundaries to create opportunities for common action. While we were with the team it was the developers who were doing the work of engaging the UX designers — they moved between floors to find the UX designers, they initiated meetings to get UX design decisions.

Within this organisational setting the interactions between the UX designers and Agile developers were localised, contingent and purposeful. What might at first appear to be a muddled set of interactions, is revealed to be the improvised, emergent course the developers were negotiating in order to progress. That is, they were interacting for particular reasons, at particular times, driven by their divisional-level commitments to getting their work done. Through the accounts and reflections of the practitioners we came to realise that in these circumstances, under their current conditions, coordination and integration of the work of the UX designers and Agile developers was achieved through the ongoing negotiations of order that served to expose the dependencies between their work.

4.5.2 UX design and Agile development work as *cooperative work*

Agile development's focus on close interactions among individuals has led to discussions in the literature emphasising *cooperation* and *collaboration*. Both are seen as significant aspects of Agile development and both are claimed to be supported by Agile methods [Abrahamsson et al. 2002; Beck and Andres 2004; Highsmith 2002; Schwaber 1995]. In the Agile literature both terms are synonymous with frequent interactions between individuals exchanging feedback concerning the software under development. However, the terms are rarely defined and we find them interchange-

ably used to refer to the interactions involving the Agile developers [Baheti et al. 2002; Qumer and Henderson-Sellers 2006], designers [Lee and McCrickard 2007], customers [Abrahamsson et al. 2003; Jokela and Abrahamsson 2004] and other development teams [Schwaber 1995].

For a clearer concept of cooperation, we turn to the area of Computer Supported Cooperative Work. The work of the UX designers and Agile developers we have discussed here has elements of *cooperative work*, defined by Schmidt:

> "...cooperative work occurs when multiple actors are required to do the work and therefore are mutually dependent in their work and must coordinate and integrate their individual activities to get the work done" [Schmidt 1994b].

Cooperative work is distinct from collaborative work in that it does not require actors working synchronously on the same task [Teasley and Roschelle 1993]. Cooperative work occurs when actors can divide the tasks between them, carry out the tasks during their individual activities and at some point fit the results together into one work product. The similarity of Schmidt's definition with the work of the UX designers and Agile developers under discussion is self-evident. Analysing their work in light of cooperative work, draws attention to the complexity of the dependencies between the designers and developers in our study, and hence, the complexities inherent in their expectations about what is acceptable cooperative behaviour.

A necessary element for cooperative work is the mutual dependence of the actors involved. Schmidt and Bannon explain that actors are mutually dependent in their work when "...one actor depends on the quality and timeliness of the work of the others and vice versa" [Schmidt 1994a]. Considering the dependencies from the developers' point of view and then from the UX designers' point of view shows that they held different *expectations* regarding their dependencies and that there was, in fact, a *lack* of mutual dependence between the Agile developers and UX designers in our study.

The Agile developers in our study were responsible for creating working software, which adhered to the UX designs handed to them, and to do so within a set deadline. Due to segregated responsibilities and the iterative, incremental rhythms of Scrum, the developers relied on the timely delivery of the UX designs, in order to make progress in their work, i.e., transforming the paper-based UX designs into working software. Thus, the *expected* dependency on the UX designers that emerges is based on the assumption that the UX designers are willing and able to accept feedback, produce a redesign every sprint based on that feedback, and will continue doing so until the release date. Scrum, and Agile development in general, is specifically designed in this way to enable the team to adapt to feedback and changing requirements. From the developers' perspective cooperation with the UX designers required an ongoing, almost turn-based effort that arises from the details of working through the implementation issues of the UX designs. This view acknowledges that issues emerge from software construction and that addressing those issues is a collective effort involving UX design as well as implementation considerations.

The evidence from our study suggests, however, that a similar dependency on the developers was not experienced by the UX designers. In contrast with the developers, the designers were engaged in work (producing paper-based wireframes and visual designs) that could progress without requiring input from the developers. We find evidence for this in the missed deadlines, surprise about developers calling a feedback meeting with the UX designers and that UX designers at no point during the observations approached the developers about design decisions. The way the UX designers worked was reinforced by the values in the organisation, i.e., that the best designs are constructed free from the constraints of implementation considerations. In this view, UX design is seen to exist independently from construction. Optimal UX design is created in a "design phase" in which UX designers are free to apply their creative energies without considerations about whether the designs can be turned into working software. It is not surprising that the UX designers in this organisation seemed unprepared to take the developers' feedback on board and redesign what they

had produced. Cooperation with the developers from the UX designers' perspective involved handing off the UX designs to the developers when the UX designers decided they had achieved the best possible designs. Emergent issues requiring further interactions with the developers would be minimal and implementation unproblematic.

4.5.3 Managing cooperation through articulation work

As a further refinement of organisational culture, Vaughan states that "rules, rituals and beliefs can evolve that are unique to work groups" [Vaughan 1996, p64]. Similarly, our analysis suggests that the Agile developers and UX designers were not wilfully acting to impede each other's progress. Rather, within their divisional groups, the developers and designers had distinct work group cultures that shaped their views on cooperation with those outside their divisions. The organisational setting promoted separation between these groups to the extent that the realities of the day-to-day work of one group was effectively "invisible" to the other — the developers could not understand why the UX designs were handed over late, and the UX designers could not understand why the developers had issues they wanted to discuss. This creates problems for considering their work as straightforward cooperative work, as the mutual dependency that would ordinarily compel cooperating actors to coordinate and integrate their work was not present in this setting. The developers, who were held accountable for producing the working software, and therefore the successful delivery of the project, had no alternative, but to elicit the kind of cooperation they required from the UX designers by means of *articulation work*.

Articulation work, defined by Strauss as "the specifics of putting together tasks, task sequences, task clusters — even aligning larger units such as lines of work and subprojects — in the service of work flow" [Strauss 1988] is evident in the work the developers were engaging in as they stepped outside their divisional boundaries to create opportunities for common action with the UX designers. The day-to-day negotiation of the dependencies and, hence, the coordination and integration of Agile

development work with UX design work was the successful outcome of articulation work. Articulation tasks are unacknowledged parts of rationalised models of UX design and Scrum development, yet without these tasks being performed, it would have been impossible for the developers to continue doing Scrum development, and thus, impossible to create working software within a deadline. Whereas articulation tasks in other contexts have been considered "workarounds" or "kludges" [Gerson and Star 1986], the articulation work performed in our study proved to be an essential part of the work of combining UX design with Agile development in practice.

4.6 Summary

This chapter presented an ethnographically-informed study of an established Scrum team in a large organisation, working with UX designers in the same organisation. The analysis shed light on the unacknowledged tasks without which it would have been impossible for the developers to continue doing Agile development, and thus, impossible to create working software within a deadline. The analysis of accounts of practice suggests that combining UX design with Agile development could be seen as a problem of combining two processes and that such a combination is how the problem should be addressed in practice. In contrast, we have shown that combining UX design with Agile development in practice is a problem situated in the organisational setting in which the UX designers and Agile developers are embedded.

Our findings are not explainable in terms of rationalised accounts of Scrum development, but in terms of trying to engage a group external to the Agile team and trying to shape them to fit the rhythm of Scrum practice. Only an examination of the values and assumptions within their organisational setting brings an understanding of how they work and why. Rather than concentrating on processes or rational plans that abstract away from the circumstances of the actions, we consider their work as situated action and emphasise that an understanding of practice requires examining the specific organisational circumstances in which the Scrum developers and UX

designers carry out their work.

By making visible (1) the concrete examples of work required to integrate UX design work with Agile development work (i.e., the coordination and integration activities), and (2) how these activities led to articulation work, we have provided the details surrounding how UX designers and Agile developers combine their work in an organisational setting.

CHAPTER 5

Study 2

5.1 Introduction to Team2A and Team2B

This chapter presents an observational study of two established Scrum teams belonging to the same organisation. Both teams had two years experience of Scrum development and had experience working with the UX designer as part of their team. The observations were conducted with both teams over six days — three days at the

start of the sprint and three days at the end. The analysis in this chapter comprises themes that emerged as common to both teams.

The next section (§ 5.2) presents background information on the people, the project they were working on and their organisational setting. Next, data collection and analysis details are discussed (§ 5.3). Next, the themes that emerged from the analysis are discussed (§ 5.4): the QA role (§ 5.4.1), sharing UX decision-making (§ 5.4.2) and fluidity of the work (§ 5.4.3). These findings are discussed (§ 5.5): by discussing the bridging role of QA (§ 5.5.2) and managing integration through awareness (§ 5.5.1). Finally, the key findings of Study 2 are summarised (§ 5.6).

5.2   Background

This study was conducted with two established Scrum teams, Team2A and Team2B, who were part of a small organisation in the UK developing for the mobile platform. At the time of the observations, they were developing software for mobile phones. The teams shared one project manager/Scrum master, but each had their own product owner, developers and Quality Assurance engineer[1]. The teams were composed of the roles as described in section 5.2.1. Whether the one UX designer in the organisation was assigned to a team, depended on the needs of the particular project. At the time of the observations, Team2A were working on a project where the client had provided the team with designs created by the client's in-house designers, whereas Team2B had the UX designer of the organisation assigned to their project.

The teams were working in the same building, spread across three floors: The project manager/Scrum master, product owners and UX designer were seated in a room on the second floor. Team2A's developers and QA were seated together in a room on the first floor, while Team2B's developers and QA were seated together in a room on the ground floor. All three floors were connected via an external fire escape and the work space of Team2A and Team2B were connected via a spiral staircase.

---

[1]Referred to as *the QA* for brevity.

Figure 5.1 shows the work space of Team2A and Team2B.

The teams' sprints were synchronised, i.e., Team2A and Team2B started and ended their sprints together. Each team began the sprint with their individual sprint planning meetings and concluded with a joint retrospective meeting. Each team held separate standup meetings at the start of each day. The sprint planning meetings and daily standups were held in their respective work spaces one after the other, so that the project manager/Scrum master could attend both.

A feature of this organisation's approach to Scrum development was their emphasis on what they called "Planning Day." This took place every second Wednesday — to conclude the sprint with a demonstration session, a retrospective meeting, and start the new sprint with the sprint planning meetings. The whole organisation gathered for the demonstration sessions and retrospective meeting at an informal venue (a co-oworking space[2]) within walking distance of the organisation's premises (see figure 5.2). The "whole organisation" constituted Team2A and Team2B, one individual in a role responsible for new business and a process administrator. It was considered to be a social (though not recreational) event where teams had the opportunity to demonstrate to the whole organisation what they had been working on, and individuals could demonstrate their gold card[3] projects. The retrospective meetings at this venue were facilitated by their process administrator, who was responsible for the creative format of the meetings. Each meeting was different, however, the format usually included breaking up into several small groups to discuss issues experienced during the sprint, and then presenting those to the larger group. A round of applause occurred at the end of every presentation.

The people in this organisation were used to doing things together. They had traveled abroad together to visit a client. The UX designer explained that during this three-week trip, they all had the chance to bond and were really comfortable

---

[2]http://wiki.coworking.info/w/page/16583831/FrontPage

[3]When the individuals in the team had completed their stories for the sprint, they were free to choose a *Gold Card* (from Higman et al. [2002, p. 105]), allowing them to spend a day working on something of their own choosing.

with each other. During one of the Planning Days, the facilitator also commented that

[2.FN.3/06.1] *[Facilitator]*: "Everyone gets on really well."

The project manager/Scrum master employed people who enjoy variation and learning in their work:

[2.FB.11/11/09.8] *[project manager/Scrum master]*: "People tend to find that their motivation and morale is lowest when we've been doing the same thing for a while."

As an organisation, they had made a conscious decision to remain a small, stable company.

Team members were assigned to teams but were not confined to those teams due to expertise. Individuals could apply their skills on any of the projects under development in the organisation at any time. This was a deliberate hiring strategy of the organisation:

[2.FB.11/11/09.8] *[project manager/Scrum master]*: "We try and avoid having silos of expertise."

When members of one team had some time to spare, they were willing to help with the work of the other team — for example, we observed the QA in Team2A helping the QA in Team2B with testing tasks. After our observations, both teams were assigned to complete the project Team2B were working on in order to complete the project on time. Based on our observations, the teams were similar in terms of work patterns and features of talk, such that the analysis resulted in similar themes across both teams. Next, the teams (§ 5.2.1)and their projects (§ 5.2.2) are introduced separately, and the common themes that emerged from the analysis are presented in section 5.4.

Figure 5.1: The image on the left shows Team2A's work space and the image on the right shows Team2B's work space with the spiral stair case connecting the two teams.

5.2.1 The people

Team2A and Team2B consisted of various roles, the titles of which were assigned by the organisation. The teams shared one project manager/Scrum master, but each had their own product owner, developers and Quality Assurance engineer (QA). Team2A and Team2B each had three developers and one QA. Team2B included one UX designer who created "screen mock-ups" which the developers could refer to as they created functional software. There was no UX designer role assigned to Team2A, as the client had their own in-house designers working for them. The project manager/Scrum master, product owners and UX designer were seated in a room on the second floor, while the developers and QA of Team2A were seated together on the first floor, and the developers and QA of Team2B were seated together on the ground floor.

5.2.2 The projects

Team2A were implementing a quiz game for the mobile phone aimed at children below reading age. The client had provided "assets" for this project, which included screen designs, wireframes, and sound and video files. The sound and video were to

be incorporated into the game. The client also provided a flash video that demonstrated how the game should work, which the team were required to follow in their implementation. Team2A were not officially required to design the user experience, as the look-and-feel and the structure of user interactions were determined by the client's in-house designers. Development time was broken up into two week sprints. When we arrived on site, the team were approximately three months into what they expected to be a four month-long project.

Team2B were developing a chat application for download on mobile phones with the Windows operating system. The client had provided a colour palette and logos for the team to use in the user interface. The UX designer had created designs that had been agreed on with the client during a workshop four months prior to our study. When we arrived on site, the team were approximately four months into the project, and was described as "nearly complete" by the project manager/Scrum master approximately four months since we concluded the observations. The team did have screen shots and a specification document to refer to. The client was based abroad.

5.3   Data collection and analysis

The observations were conducted at two different stages of the sprint. The first period of observations was conducted during the first three days of the sprint and the second period covered the last three days of that sprint. Figure 5.3 shows the timing of the observations with the sprint of Team2A and Team2B. Work activities, and therefore the observations, occurred at two different venues. Observations of the retrospective meetings took place at the informal venue, while the rest of the observations were conducted in their offices. An equal amount of time was spent observing each team. Each team was observed separately for two working days — one day at the beginning of the sprint and one day at the end — and jointly on the planning days. The UX designer was away for the last three days of the observations.

During the first period of observations, a semi-structured one-to-one interview was conducted with the UX designer in a separate meeting space, at an agreed time, with questions that were based on what had been observed up to that point. This interview was recorded via hand-written notes and transcribed into electronic form. The interview gained more background on the UX designer and his view on his interactions with the developers. During the interview, for example, the UX designer's account supported the observed ongoing creation of awareness between the developers and UX designers. The transcription recorded the UX designer as mentioning that

> [2.IN.4/6/09.1,2] *[UX designer]*: "... if developers come up with issues, these are discussed on a case-by-case basis as they come up ... There are lots of small things which are easily worked out on a day-to-day basis."

Data was collected via observations of daily work and engaging with the participants in discussions about documents, records and other tools used in their work. The researcher had access to Team2B's Skype chats, which were copied and pasted into an electronic document for later reference. The Skype chats covered the online conversations over the two weeks of the sprint and supported the themes from the analysis of the field notes generated by the observations. For example, the discussions involving the QA shed light on the role of QA in Team2B, his involvement in decision-making and raising testing issues. Some artefacts were shown to the researcher, e.g. some of the wireframes and outputs from the retrospective meetings. Other artefacts were only discussed, e.g. their project management software. The detail of practice was documented via contemporaneous field notes and photographs/sketches of the physical layout of the work area.

5.4   Findings

In Study 1, work activities were clearly visible and observable, such that themes centred around the types of activities that allowed the Agile developers to transform the paper-based UX designs into working software. The separation of the Agile develop-

Figure 5.2: Both images are of the informal venue where the teams held their retrospective meetings.
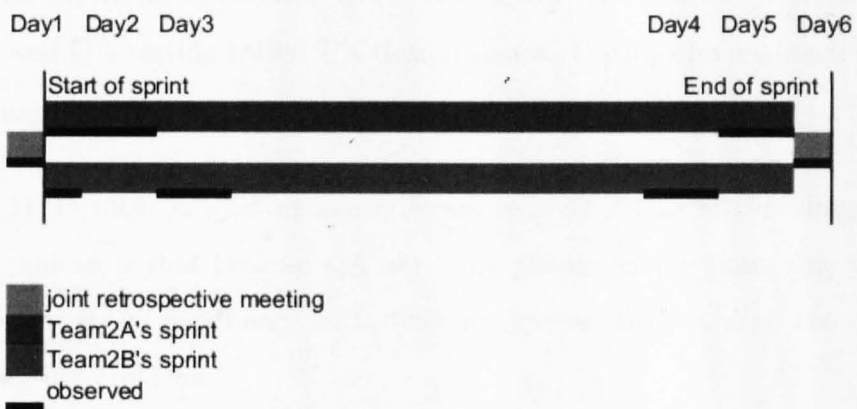
Figure 5.3: An image showing the timing of the observations with the sprint.

ers and UX designers made these themes crucial in understanding how the developers and designers combined their work in Study 1.

In this study, the environment in which the teams were working, the team set-up as well as the platform they were developing for, were different. Thus UX design and Agile development issues arose differently. This required a shift for the researcher from concentrating on tasks or activities, to focusing on talk and interactions between individuals. With the teams in this study, there were no distinct and visible UX/Agile integration activities observed. Instead, it was observed how all roles on the project were taking part in frequent discussions, creating awareness and sharing decision-making responsibilities. Conversations spontaneously occurred during Scrum activities, as well as resulting from testing activities. The relaxed interactions of Team2A and Team2B were in contrast with the systematically planned activities that pre-empted the conversations between the developers and UX designers of Team1. Next, the themes arising from the analysis are presented (§§ 5.4.1–5.4.3).

5.4.1 The QA role

The QA role featured prominently in discussions around UX design-related issues that arose during the observations. QA feedback was valued during UX design-related discussions and QA testing raised UX design issues. During the feedback session, the project manager/Scrum master commented on the role of QA:

> [2.FB.11/11/09.4] *[project manager/Scrum master]*: "One of the things that we've noticed is that because QA are using phones all the time, they've got a wide variety of experience with different devices, they're often the people noticing UX problems."

This role was required to define and carry out acceptance tests on the software created by the developers, using various mobile devices. In Agile teams acceptance criteria are ideally specified by working closely with the customer[4]. In this organisation,

---

[4]For example, http://www.extremeprogramming.org/rules/functionaltests.html.

specifying acceptance criteria depended on the QAs and their consultations with the developers — not something the QAs were comfortable with. In a retrospective meeting we attended, the QAs mentioned that they wanted "better" and "cleaner" acceptance criteria that are defined by the client, rather than the development team.

The QAs felt they were working well with the developers — describing their relationship with the developers as "healthy" — which is in line with recommendations that Agile developers and QA should be working closely together[5]. However, in this organisation, they viewed the QA role as still evolving. The data contains instances of how the participants were still figuring out how to manage the workload for the role. According to the project manager/Scrum master, their experience was for QA to go from having nothing to do to being extremely stressed in a sprint. During the sprint planning meeting, they were actively trying to spread the workload more evenly throughout the sprint. They reviewed the issue in the retrospective meeting and discussed the possibilities of writing acceptance tests as soon as developers start coding. The workload issue was not resolved during the time of this study. The QAs themselves were negotiating their workload and what their role should be during the normal course of the work. In the Skype transcript of Team2B, the QA chats with the project manager/Scrum master about documenting test cases and sharing those test cases with a client:

> [Mon Jun 15 2009 12:21:37] *[project manager/Scrum master]*: Well... the doc is a list of test cases. If we can automate some of these: great. If not we have to check them at least when the story relating to them is finished, and then again before final delivery. Ideally the more we can test regularly (thru a combination of auto and manual testing) the better tho.
>
> [Mon Jun 15 2009 12:21:52] *[QA]*: Sure
>
> **[Mon Jun 15 2009 12:22:22]** *[QA]*: **I'm happy to share the cases with them, less so about comitting [sic] myself to going through them on**

---

[5]For example, http://www.extremeprogramming.org/rules/functionaltests.html.

a regular basis as the number of them will quickly add up as we do stories...

[Mon Jun 15 2009 12:22:57] *[project manager/Scrum master]*: Right, I think that's somewhere where *[Team2A's QA]* will have to help out then. I don't think I could argue that we'll write test cases but not recheck them.

The above excerpt appears in the sequence of Skype chats about different versions of the *Sun Java Wireless Toolkit*[6] and the manager asking for a response to a question from the client on a different matter. The place in the Skype discussion in which the excerpt appears illustrates not only what was negotiated, but also the place of the negotiations in the course of the day.

The QA role in this team spanned developer- as well as designer-related interests. Through the QA's acceptance tests the user experience was evaluated by matching expected behaviours (as intended via the design by the UX designer) with the actual behaviour of the software (as implemented by the developers). Although the QA was not expected to create the user experience of the software, the QA's testing of the software highlighted issues that would have affected the user's experience had they not been addressed. During the feedback session the manager commented:

[2.FB.11/11/09.4] *[Project manager/scrum master]*: "Certainly I'd see QA as testing not the code but the whole product."

On a day we were not present, the Skype transcript captured a conversation in which the QA raised that on a certain model of handset the terms and conditions were not being displayed to the user. This alerted the team to a possible difference between their old and new builds. During the follow-on exchanges they decided to investigate which build caused the problem and determine who was responsible for fixing it (the client or the team):

[Tue Jun 9 2009 11:44:09] *[QA]*: Z10 can't get T&Cs...

---

[6]For example, http://www.oracle.com/technetwork/java/index-jsp-137162.html

[Tue Jun 9 2009 11:44:50] *[project manager/scrum master]*: did it not used to work?

:

[Tue Jun 9 2009 11:45:37] *[QA]*: It used to work...

:

[Tue Jun 9 2009 11:48:55] *[product owner]*: i got that error on my n95 on a recent build

[Tue Jun 9 2009 11:49:00] ...but it had worked before

[Tue Jun 9 2009 11:49:07] *[developer]*: o rly

[Tue Jun 9 2009 11:49:46] ...perhaps something in the new MSP stack is messing us up

[Tue Jun 9 2009 11:49:49] *[QA]*: that's a point

[Tue Jun 9 2009 11:49:54] *[developer]*: *[QA]* can you go back and try an old build

:

[Tue Jun 9 2009 11:52:56] *[developer]*: whose responsibility is that

:

[Tue Jun 9 2009 12:05:23] *[project manager/scrum master]*: Can we compare the sample MIDlet from their new MSP code to the one on the old MSP code? If the new one breaks and the old one works...the problem was theirs [the client's].

The QA was observed to be "championing" both the end user and developer perspectives when the acceptance tests pointed to problems with the software. In a standup meeting, in response to hearing that an advertisement appears as a chat, the QA asked:

[2.FN.5/06.7] *[QA]*: "What users are going to say yes I want ads?"

In the same meeting during a discussion about missing specific information from the client, the QA mentioned that he was

> [2.FN.5/06.6] *[QA]*: "... wary of starting stories without behaviours which means we might have to change things and that wastes time."

The QA was likely aware that what he defined as acceptance tests determined the work of the developers. He was careful about making assumptions about how users would use the software. In the retrospective meeting he raised this concern, suggesting that the organisation demand more detailed acceptance criteria for a story. At the time, the UX designer as well as one of the developers supported the idea of showing the client working software and adapting the software based on their feedback. However, the QA was uncomfortable with this:

> [2.FN.16/06.15] *[QA]*: "They tell us what they want and we make it, not make something and tell them this is what they're having."

5.4.2   Sharing UX decision-making

In Team1, UX design issues were the sole province of the UX designer. That is, only the UX designers could make UX design decisions and it was left to the developers to approach the UX designers when issues arose while transforming the paper-based designs into working software. Team2A and Team2B handled UX design issues in a markedly different manner from that of Team1. We observed how decision-making was shared among the members of Team2A and Team2B. Every role on the team shared concern for making progress not only with the UX design work but also with the implementation work. Team2A did not have a UX designer role as part of their team, however, UX design issues emerged during their daily work and the team made decisions that affected the user experience. For example, decisions relating to how a UI element specified in the video from the client should be adapted for the screen of the mobile phone.

As UX design issues emerged in their work, these were dealt with by the team in a way that suggested changes to the UX design were unremarkable. The UX design issues were mentioned alongside non-UX issues during the course of the day, discussed among the team, and a decision made to either implement a solution or pass the issue on to the client. During a standup meeting with Team2A, one of the developers informed the rest of the team that he had replaced some visual designs. When the researcher probed for further details regarding this change, the developer's brief explanation ended with "So that's all that was." Giving the impression that their changes and additions were in no way strange. The developers mentioned that the client was aware that the game that the team delivered will not be exactly as they specified — the developers commented that it was unnecessary to go back to the client with every decision. Their implementation of the UX design was further justified when the team expressed that nobody had noticed anything "wrong" when the software was demonstrated to the others in the organisation. It was clear that throughout the project, they were making small decisions to adapt the client's flash-based examples to the mobile phone platform, and they could do so because of an established understanding between the team and their client, and a reliance on others in the organisation to notice problems as the software was demonstrated.

The above examples come from our observations with Team2A, however, UX design issues were handled similarly in Team2B: UX design issues were mentioned alongside non-UX issues during the course of the day, everyone in the team had input, which led to a decision being implemented or the issue being passed on to the client. The developers referred to the UX designer for certain decisions, for example, how an advertisement appears in the application, or whether a UI element is highlighted. Other decisions were arrived at as a team, in a way that made it difficult for the observer to pinpoint how the input of the UX designer resulted in the outcome of the decision. For example as the team learned more about how the software functioned on the mobile platform, a situation emerged for which they did not have specifications from the client. There were two options — one in which the upgrade was mandatory

and the other in which the user could disallow it — they decided on implementing both versions. The problem was not handled specifically as a problem in the domain of UX design. The decision to implement two versions emerged from the collective discussions, and then implemented by the developers. The UX designer confirmed in the interview that the developers' closeness to the implementation led them to occasionally come up with better UX design decisions than he could. Their focus on getting the job done, relied on the frequent conversations about the work as they went and an established understanding that team members' input into UX design issues was valuable, irrespective of their role.

Sharing responsibility for UX decision-making was observed to be congruent with their openness to sharing information within their teams, e.g. via regular status updates on tasks, and also with members of the other team. They walked to the work space of the other team and started conversations about happenings with their project, e.g. that the client had sent the team the wrong sound file. The creation of awareness of each other's work was further supported by the shared demonstration and retrospective sessions held at the end of every sprint, so that relying on other team members as a source of feedback was a credible course of action — as Team2A had relied on the others at the demonstration session.

5.4.3 Fluidity: UX/non-UX issues look the same

In contrast with what was observed in Study 1, in this study, all roles on each project were observed to be taking part in frequent discussions, creating awareness and sharing decision-making responsibilities. During the course of the day, UX design issues were generated and dealt with alongside technical implementation issues. In a similar way to Study 1, the UX design work was broken up into stories and prioritised along with the rest of the backlog. However, in contrast with Study 1, breaking up and prioritising the UX design work was no more remarkable to the teams than carding and prioritising the non-UX related work. We observed that the UX designer was

continually aware of the implementation issues of their designs and the developers were constantly aware of (and participated in) UX design decisions. Every role on the project was involved in discussions that promoted consideration of design aims alongside technical constraints. The effect was that UX design issues were inseparable from implementation issues.

Dependencies between the UX designer and developers in Team2A and Team2B were not explicit, or intended to be well-defined. The manager hired individuals who could fulfill the responsibilities of their main role, but who were also able to contribute outside their roles, or who could learn to do that. In contrast with Team1, we observed that the individuals in Team2A and Team2B were comfortable making decisions and giving advice to other roles. For example, when the QA asked a question regarding a test case during a standup meeting, the UX designer replied that it was an "extreme" edge case and therefore not a priority for testing. Everyone agreed and the test case was discarded.

In Team1, the developers were dependent on the UX designers for making design decisions because it was understood that decisions were made only within their disciplines (developers develop and designers design). In Team2A and Team2B, dependencies between roles appeared fluid. This was observed between the developers and UX designer in their interactions with each other and their decision-making. The developers engaged with the UX designer based on a tacit understanding of when UX design input was required. The developers knew when decisions were appropriate to be made without the UX designer and when not. In response, the UX designer made himself available to be approached at any time. During the sprint planning meeting, the UX designer was not present to start with and when an issue was brought up, they decided to fetch the UX designer. In an interview with the UX designer, he explained that although they work together through lots of issues, the issues are generally small and "easily worked out on a day-to-day basis." The UX designer relied on the developers to ask for input as required and the work of the developers determined when that input was required. This was how the team were comfortable

engaging, which can be attributed to a mutual awareness of what was required that arose from their close interactions as a bonded team. Fluidity understood in terms of how they managed dependencies between roles.

5.5 Discussion

As with Study 1, a picture emerges concerning work arrangements, dependencies and mechanisms that make the integration and coordination of work possible. The account in this chapter emphasises the localised, contingent and purposeful work carried out by the developers and designer.

Team2A and Team2B seem rather similar to Team1: these teams were embedded in organisations that recognised the combination of UX design and Agile development (Scrum) as essential for software development, and had the UX designers sitting on separate floors to the developers. Despite these similarities their experiences of practice were quite different. Team1's systematic, separatist approach to integrating and coordinating UX design and Agile development contrasts with Team2A and Team2B's subtle ongoing effort that promoted shared awareness of UX design aims and technical constraints, as well as shared decision-making. In the case of Team1, the developers were expected to take direction from the UX designers whereas in the case of Team2A and Team2B the various roles were negotiating what that direction should be. The prominent place of the QA role in their integration of UX design and Agile development.

5.5.1 Managing integration through awareness

In this study, we have been concerned with the coordination of Agile development work with UX design work as the moment-by-moment negotiated dependencies, as negotiated by those involved in the daily work, rather than a coordination process imposed on the team from outside. Influences external to the teams evidently shaped their work and dependencies in their work, however, the details of the workflow were

seen to be negotiated between the individuals in the teams, in the spirit of self-organisation.

In the previous study, the organisational setting of Team1 promoted separation between the UX designers and Agile developers to the extent that the realities of the day-to-day work of one group was effectively invisible to the other — the developers could not understand why the UX designs were handed over late, and the UX designers could not understand why the developers had issues they wanted to discuss. In Team1 the developers elicited the kind of cooperation they required from the UX designers by means of articulation work. Coordination efforts in Team1 were reliant on explicit articulation work that was aimed at engaging the UX designers.

In this study, the coordination of UX design and Agile development work was more implicit than in Team1, that is, coordination arose from the ongoing interactions within the teams. Rico et al. [2008] distinguish between *implicit coordination* and *explicit coordination*. Implicit coordination is achieved as follows: "...by anticipating what others in the team are likely to do, members can adapt their own behaviour to facilitate the team's task completion without explicit discussion of who should do what." Team2A and Team2B's implicit articulation work, and hence their implicit coordination, is evidenced by the difficulty of disentangling UX design issues from Agile development issues, the unremarkable enactment of shared responsibility for decision-making, and their fluidly negotiated dependencies.

As noted in section 5.4.3, the developers engaged with the UX designer based on a tacit understanding of when UX design input was required. The developers knew when decisions were appropriate to be made without the UX designer and when to refer to the UX designer. Therefore, as well as coordinating activities, Team2A and Team2B were also coordinating expertise, as Faraj and Sproull [2000] explain: "...teams must be able to manage their skill and knowledge interdependencies effectively through expertise coordination, which entails knowing where expertise is located, knowing where expertise is needed, and bringing needed expertise to bear."

Coordination as enacted by the teams in this study, relied on an *awareness* of each others' skills, availability, etc. Indeed, awareness between team members and the tools and practices that support it [Whitworth and Biddle 2007] is a prominent theme in Agile development (e.g. awareness of project status supported by information radiators [Cockburn 2004, p. 54], awareness of what aids or hinders the team's success [Beck and Andres 2004, p. 6], awareness of others and their availability for answering questions [Cockburn 2006, ch. 5]). The practices of the teams in this study, allowed creation of a shared awareness that manifested as fluid interactions and shared decision-making. The shared awareness gave rise to implicit coordination of Agile and UX work and their ability to self-organise.

5.5.2 QA bridging between developers and the designer

The complex role of QA in this study emerged from how roles and the dependencies between them were negotiated. In this study, "getting the job done" was achieved via a blurring of roles and responsibilities such that individuals were flexibly applying their skills as required. Therefore, the nature of the QA role was shaped within the context of the fluid boundaries between roles, the dynamic, frequent interactions between individuals in the teams and an evolving understanding of what the role required. However, there was an uneasiness around the QA role that came from the lack of specific acceptance criteria from the client. The QAs were not convinced that what they were specifying for acceptance criteria were what the client actually wanted. Despite not being entirely comfortable with the situation, QA was getting the work done. As a bridging role then, QA was bridging between certainty and uncertainty by specifying acceptance tests. The acceptance tests became the concrete criteria to which the implemented software could be compared.

In this study, QA clearly had a liaising role between the developers and designer. The QA roles of Team2A and Team2B were instrumental in maintaining the mutual awareness between the developers and designer, by generating opportunities for

discussions of UX design issues. In Team2A, the test results generated by QA highlighted UX design issues that could be resolved either through discussion between the team, or through contacting the client. In Team2B the test results that the QA generated raised issues that then ensued in conversations between the developers and designer. The QA role was clearly a kind of intermediary role enhancing collaboration between developer and designer-related issues and bridging between Agile development and UX design. This was a carefully negotiated process that depended on the tacit understanding of each other's abilities and expectations.

Agile development teams have explicit descriptions for the required roles, for example, the *product owner, scrum master* and *team* of Scrum [Schwaber 2004, Chapters 3, 5, 8], and the *coach, tracker, programmer, customer, tester, consultant* and *big boss* of Extreme Programming [Beck 1999]. However, the literature contains evidence that these acknowledged roles are successful when intermediary, less acknowledged, roles exist to liaise between them. Liaising roles, or bridging roles, facilitate and enhance the collaboration that Agile software development depends on. Martin [2009]'s PhD research on the customer role identified three *Collaboration Guides* that were found to "enhance the business-technical collaboration on a project." Hoda [2011]'s PhD research on self-organising Agile teams identified a similar role as the *Translator*, that "translates between the business language used by customers and the technical terminology used by the team." Rising [2003] describes how the *Product Champion* builds "a bridge of trust between the development team and the customer." Abrahamsson et al. [2010] ask whether Agile development and software architecture can co-exist and point out that the gap between the methodologies are bridged by the developers and architects.

Although the analysis in Study 1 did not highlight a bridging role between the developers and designers, the effect was also recognisable there. At the time of the observations, the client-side developers and the product manager/product owner were identified as being the roles that interfaced most frequently with the UX designers (§ 4.3). They were therefore selected for semi-structured interviews in Study 1.

5.6 Summary

This chapter presented an ethnographically-informed study of two established Scrum teams in a small organisation, working with one UX designer in the same organisation. For the researcher, the nature of the work the developers and designer engaged in, required a shift away from trying to identify observable integration activities, to analysing talk and interactions. With this focus, the analysis shed light on how all roles on the project were taking part in frequent discussions, creating awareness and sharing decision-making responsibilities. Team2A and Team2B's implicit articulation work, and hence their implicit coordination, is evidenced by the difficulty of disentangling UX design issues from Agile development issues, the unremarkable enactment of shared responsibility for decision-making, and their fluidly negotiated dependencies. Everyone on the team's input into UX design issues was valued, irrespective of their role in the team. The QA role featured prominently in UX design-related issues, championing both the end user and developer perspectives. QA testing ensued in discussions about UX design issues between the developers and designer. The QA was therefore instrumental in the mutual awareness that was maintained between the developers and designer, effectively bridging Agile development and UX design issues.

CHAPTER 6

Study 3

6.1   Introduction to Team3

Unlike the teams in Studies 1 and 2, the team in Study 3 was not an established
Scrum team. Indeed only one individual had any experience of working as part of
an Agile team. Team3 had chosen to trial Scrum with one of their projects and, as

part of this trial, to seat the developer and designers in the same room for two days. This chapter presents an ethnographically-informed study with the developers and designers doing Scrum development, spanning two work days.

The next section (§ 6.2) presents background information on the people, the project they were working on and their organisational setting. Next, data collection and analysis details are discussed (§ 6.3). Next, the themes that emerged from the analysis are discussed (§ 6.4): how the team performed Scrum development and negotiated progress (§ 6.4.1), decision-making and common action (§ 6.4.2), and engaging and shaping dependencies (§ 6.4.3). These findings are discussed (§ 6.5): discussing how the team were (re)creating progress (§ 6.5.1) and expectations about acceptable behaviour (work group culture) (§ 6.5.2). Finally, the key findings of Study 3 are summarised (§ 6.6).

6.2   Background

Team3 was part of a Digital Media Agency in the UK that applied User Centred Design (UCD) on all of their projects. As an organisation they were experienced with using waterfall-based PRINCE 2 processes, but, according to the Head of User Experience (UX), changing client needs had signalled that it was time to explore something new. The Head of UX therefore proposed they trial Scrum for the duration of a complete project. Within this trial, the Head of UX also proposed that the developer and designers sit together in one room for some of the time, specifically with the aim to bring design and development closer. The idea had come from outsourcing one of their information architects (IA) to a client organisation who had asked that the IA sit and work in the same room with a developer and designer at the client premises. The Head of UX explained that

> "Talking to the IA about his experience at the client's premises prompted us
> to consider whether doing the same could also make our process easier and
> quicker," [Tzanidou and Ferreira 2010].

One room that was normally used as a meeting room or as a laboratory for user evaluation studies, was booked for Team3 to sit together (see figure 6.1). Due to organisational constraints they could only book the room for two days, even though the project was expected to last several weeks[1]. These were the two days that we observed the team working. People who walked past the room would comment about the team being in the room. The technical support person commented: "Is this the Agile room?" and: "Are you Agile?" While this is playful it also shows that the team was doing something different that impacted beyond the immediate team:

> [3.FN.1.24] *[Technical support]* comes in and says "Are you Agile? The team replies: "Extremely Agile."

The developer and designers in this study worked in a building with four levels where the third was taken up by another organisation. The UX designers normally worked on a separate floor to the developers. According to one of the UX designers on Team3, the designers and developers had little interaction and rarely talked to one another. Following their waterfall approach, the designers created all the wireframes and visual designs for a project, and then handed those to the developers to implement. The developer on Team3 mentioned that the developers rarely go back to the designers when they run into issues:

> [3.FN.2.2] *[developer]*: We often find things that don't work, but we don't go back to design. We just do our best.

Therefore, much of what they did during the two days they were sitting together was "new" to them. This team frequently reflected on how the way they were working in this trial differed with how things would "normally" work.

This was the organisation's first attempt at Scrum development. It also served as an opportunity for the organisation to get hands-on experience with Scrum practices and to learn how Scrum would work with their existing organisational structures.

---

[1]The last time we visited the team, they had completed five cycles and estimated that they had two cycles left until completion.
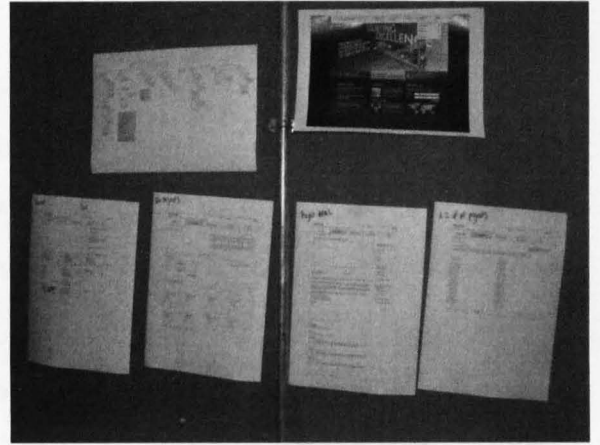
Figure 6.1: The image on the left shows how the team was seated in the "Agile Room" and the image on the right shows some wireframes fixed to one of the walls.

Before this point, the Head of UX had spent a year bringing awareness of Agile methods to the organisation and building up trust and enthusiasm for a change.

6.2.1 The people

Team3 consisted of one project manager, one developer, an information architect (who created the wireframes[2] and had no previous Agile experience) and one visual designer (who created the visual design[3] and had some previous Agile experience). We consider the visual designer and information architect as UX design roles. The developer was responsible for providing the back end code for the front-end designs. During the time of the observations, the developer was working on setting up a Content Management System for the client website and prototyping ideas to test whether a design could be implemented. This was their first experience as a Scrum team. The work for this project was contained within the team, i.e, Team3 were independent of the work of any other team and no other team depended on their work.

---

[2]By *wireframe* we mean the popular design tool, which conveys content and layout information, as described by Garrett [2002]. Similarly to the interaction designer in Study 1, the information architect in this study created wireframes as representations of the website under construction.

[3]Similarly to the graphic designer in Study 1, the visual designer created the *visual design* as a representation of the website under construction, conveying the content and layout information along with graphical aspects of the design.

6.2.2 The project

The project chosen for this trial was redesigning a client website. At the start of the project the client had handed over results of a focus group that they had held concerning their existing website. Team3 were expected to design the new website incorporating these results. Shortly after the project began, the client reduced their initial budget by two-thirds, although they still wanted a high impact website. Resources for this project were limited and had to be carefully managed. The Project Manager explained: "The client paid a fixed price, so we have to get as much as possible done for that price." The client was prepared to allow the creative solution to emerge and had agreed to be heavily involved. Team3 were communicating with the client by attending scheduled meetings, posting messages online, or via the Project Manager, who forwarded their questions on to the client.

The project was to be completed in cycles[4] set to be one week long. At the end of cycle 1 the client had rejected the designs that had been shown to them — the designs were considered not corporate enough and the client wanted more emphasis on the "Who we are", "Our projects" and "Working for us" parts of the website. We observed Team3 for two consecutive working days, which coincided with the first two days of cycle 2.

6.3 Data collection and analysis

The observations were conducted on the first two days of the team's second cycle. Figure 6.2 shows the timing of the observations with the cycle of Team3. Work was carried out in the "Agile Room" with the UX designers and developer present on both days and, therefore, the work was observed in this room. No semi-structured interviews were conducted with individuals, as all roles involved were observed throughout the two work days. Questions based on the observations were presented to the team

---

[4] *Cycle* is used here due to the organisation in this study using this terminology. It is equivalent to *sprint* in Scrum terminology.

Day1 Day2

Start of cycle   End of cycle

Team3's cycle
observed
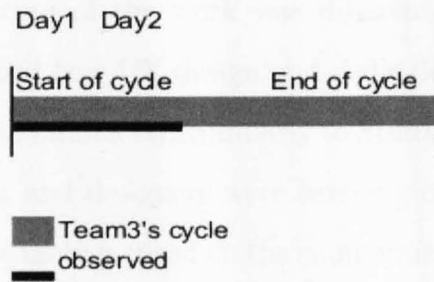
Figure 6.2: An image showing the timing of the observations with Team3's week-long cycle.

in the feedback session and the group discussed their answers together. This part of the feedback session was recorded and transcribed.

Questions presented to the team in the feedback session gained more information on the current status of the project, their view on their Agile practices, how working together has changed since the two day trial of sitting together, and their biggest challenges faced so far. The importance the team placed on the immediacy with which changes can be dealt with when sitting together, was emphasised during this discussion. For example, the developer mentioned that "the advantage is as soon as [the IA] changes it, changes her mind, I can do it instantaneously."

Data was collected via observations of daily work and engaging with the participants in discussions about documents, records and other tools used in their work. Some artefacts were shown to the researcher, e.g. some of the wireframes, the visual design and the repository where they kept electronic documents and discussions with the client. Other artefacts were only discussed, e.g. the Content Management System. The detail of practice was documented via contemporaneous field notes and photographs/sketches of the physical layout of the work area.

6.4 Findings

In Study 2, there were no distinct and visible UX/Agile integration activities to observe. All roles on the project were taking part in frequent, spontaneous discussions, creating awareness and sharing decision-making responsibilities.

In this study, the nature of the work was different, the seating arrangements were different, which shaped how UX design and Agile development issues arose and where the focus of the observations lay. Similarly to Study 2, the researcher was aware of the tasks the developer and designers were carrying out during the observations, however, attention became more focused on the communication between the developer and designers. Their sitting together had consequences for their communications and, therefore, how they made progress in their work. The emergent themes attend to features of work (workflow) and making progress. The themes are presented in sections 6.4.1 6.4.3.

6.4.1 Scrum development and negotiating progress

This team referred to themselves as a Scrum team. The development work was broken up into cycles, team members were briefing each other on what they were going to work on at the start of the day, there was evidence of regular client involvement, and requirements were expected to emerge. Agile characteristics were evident in team members keeping each other informed of their progress, sharing knowledge and self-organising as a team. The rhythm of the day was shared by the developer and designers, and consisted of coming into the office in the morning, breaking for lunch and leaving at the end of the day. There were no commitments to other rhythms outside the team (for the two days the designers and developer were sitting together). The developer and designers were committed only to maintaining a continuous flow of work and making progress towards the wireframes and visual designs they agreed to show to the client at the end of the cycle.

At the start of the project the client had handed over results of a focus group that they had held concerning their existing website. Team3 were expected to design the new website incorporating these results. This was an unusual situation for the team to be in. As the Head of UX explained:

[3.FB.27/07/09.1] *[Head of UX]*: "And when we first met with the client, the

first thing they did was to present to us the findings of that initial research. So we can then go and propose what would be the next step, which is quite different from any other type of work we've been invited to get involved in."

No stories or list of features for a product backlog was created with the client. Instead, the client expected the solution to emerge during the project cycles and requirements to be refined after viewing the working software.

After the first cycle, the client provided feedback on the first set of designs: The designs were considered not corporate enough and the aspects of the website the client had wanted highlighted were not prominent enough. So in the second cycle, the designers were working to incorporate that feedback into their designs. As the client was expecting the requirements to emerge during the project cycles, there had been no official sign-off on the designs and Team3 felt the client requirements were still very vague. This resulted in a lot of uncertainty concerning the "vision" for the website and what they were aiming towards. One exchange between the developer and designers in the fieldnotes is noted as follows:

[3.FN.1.24] *[visual designer]*: I just feel like I'm not getting anywhere.

[IA]: I also feel that way.

[developer]: My stuff's working but I don't know if what I'm working on is going to be used.

The visual designer was particularly uneasy about the client not signing off on designs, asking:

[3.FN.1.4] *[visual designer]*: "How will that work? How will we know how to start? Something needs to be signed off."

However, we saw the team having conversations and moving on in their work. Rather than coming to a standstill they were **negotiating progress** as a team. In addition to the focus group results, Team3 were discussing what needed to be done

by referring to meetings with the client, the existing website and some information on the client organisation (e.g. management structures). Among themselves the team were also deciding on what it meant to progress. For example, they had no burndown chart, or wall of stories to indicate that work was being completed. To ensure they continue the developer suggested they base their designs on the existing website:

[3.FN.1.9] *[developer]*: "Just cover the site map."

The developer and designers were negotiating the way forward by asking and answering questions, keeping each other informed of decisions, and involving each other in decision-making. They were moving forward by

- resolving questions such as, "What do you remember from the client meeting?" "What did you understand the client wanting from the client meeting?"

- informing the others of how they plan to approach the task, for example, "I'm going to make two versions of this website."

- involving the developer in design decisions. For example, in a discussion about exploring alternatives to a drop-down list, the UX designer asked the developer directly whether her idea would be more difficult to implement. The developer immediately gave her an answer:

[3.FN.1.15] *[developer]*: "It would probably be just as easy."

The visual designer commented that sitting together meant everyone could collaborate in order to get their points across, rather than having the designers "dictate" due to lack of communication. The feelings about sitting together in this new arrangement were very positive — the team commented that sitting together allowed better focus due to fewer distractions, gave them opportunities to ask each other questions and gave them all access to "content" at the same time. Sitting together enabled the team to negotiate progress and, therefore, the direction of the software together.

6.4.2 Decision-making and common action

By *common action*, we mean the work the developers and designers do collectively when they are focusing on the same part of the software together (synchronously). For example, the discussion that ensues when either the developer or designer raises an issue and calls everyone's attention to a specific design element (e.g., a text box on a web form) or client requirement. Common action usually occurred during face-to-face discussions. In Study 1, common action between the developers and designers was necessarily pre-empted with searching for a UX designer, sending emails, and/or organising a formal meeting.

In this study, common action was an informal part of the workflow. Decisions that the developer and designers needed to make together, were made without delay. Figure 6.3 is a characterisation of the type of talk that initiated opportunities for common action. The figure shows two types of common action — that between the designers themselves and that between the developer and the designers. It is important to note that although the developer did not participate verbally in the common action of the designers, he was present during the exchanges and therefore could have interjected if he had felt it was necessary to do that. We know that being present during the designers' exchanges was a valuable part of the developer being aware of design decisions. In the feedback meeting he mentioned that

> [3.FB.27/07/09.2,3] *[developer]*: "... looking through *[the IA]*'s work in progress and also *[the visual designer]*'s work in progress to work out if I could see any problem areas and that was really valuable doing that... the advantage is as soon as *[the IA]* changes it, changes her mind, I can do it instantaneously."

Figure 6.3 shows that the designers asked the developer for feedback about implementation possibilities and client requirements, to which the developer responded. For example, the IA asked the developer about search options on the web page and whether one was more difficult to implement than another. The developer responded saying that both were possible and easy to do. There were also examples where the

designers asked the developer about what he remembered about client requirements, based on a client meeting that occurred a few days before the observations:

> [3.FN.1.24] The IA asked the developer a question about how many projects the client said they would want to add to the website and the developer responded that it would be a few every week.

When the developer initiated common action it was by giving suggestions for design elements   he suggested they design an "interactive roll-over icon" in order to make some information on the website more prominent   and anticipating upcoming design issues:

> [3.FN.2.14,15] *[developer]* asked the IA whether there was "anything special about the search page?" to which she responded with a brief explanation of how results should be filtered.

Vital to the team progressing and "getting on with work" were the instances of common action where opinions, advice and other pieces of information could be shared. The questions that initiated common action were important examples of articulation work that drew the team's attention to aspects of the software under development that required more discussion. Common action helped the developer and designers coordinate their work and decide what further action was meaningful with respect to the software.

6.4.3   Engaging and shaping dependencies

The Head of UX explained in a separate conversation that the designers should take the lead in this project:

> [3.FN.1.19] *[Head of UX]*: The project was set up so that the designers would take the lead.

Dependencies between the developer and designers were therefore set up before they started work. During the two days we observed, the designers were working ahead
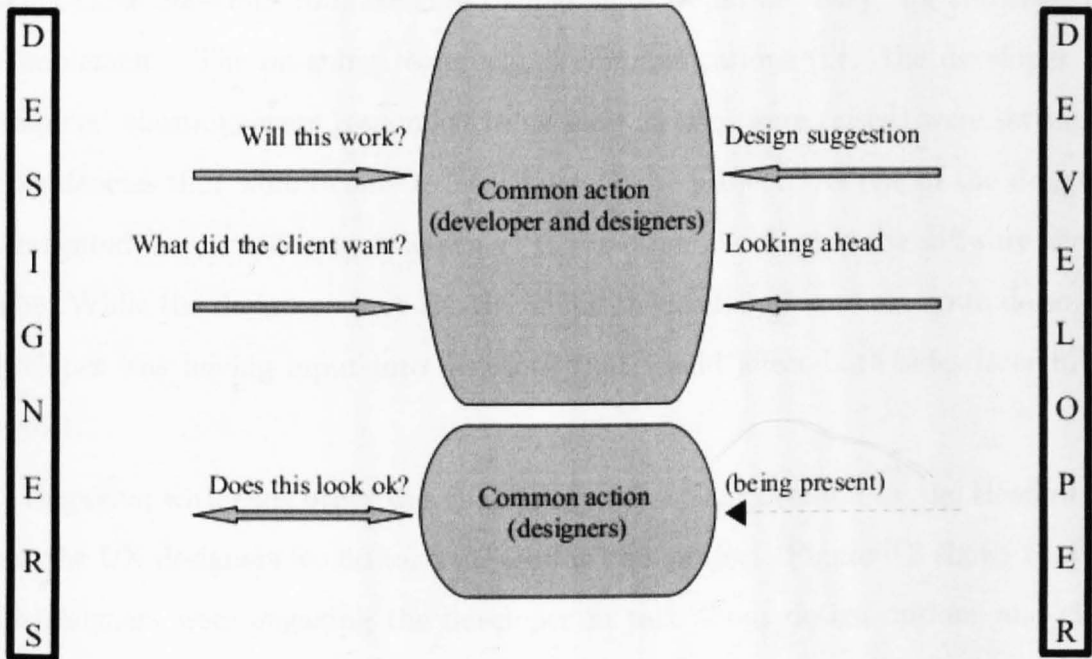
Figure 6.3: Talk in Team3.

of the developer — they were creating wireframes and visual designs for the website, while the developer was working on setting up the Content Management System. Apart from mocking up some prototypes to test whether an idea would work, the developer was not turning the designs into working software during those two days. In light of this, we observed that the developer was not asking the same kinds of questions as the developers in Study 1 — he was not running into implementation problems. Similarly, the designers were not distracted with other projects (they were involved with other projects, but were giving priority to the trial) and the designs were still in the early stages.

During these two days, the developer and designers engaging with one another can be seen as shaping their subsequent dependencies on one another. For example, the designers asking the developer a question about what he preferred to implement, was a way of mitigating the risk of the developer running into designs that could not be turned into code later in the project, and the extra effort that would be required in that case. The designers explicitly told the developer that they were

asking these questions to make sure their designs would be "easy" for the developer to implement. The on-going reciprocated communications (i.e. the developer and designers' questions were responded to as soon as they were raised) were setting up dependencies that would come to bear later in the project. As one of the designers commented, no one side was "dictating" to the other about what the software needed to be. While the designers were clearly taking the lead, and were set up to do so, the developer was having input into decisions that would affect both sides later in the project.

Engaging with each other was shaped by the explicit decision by the Head of UX that the UX designers would take the lead in this project. Figure 6.3 shows that the UX designers were engaging the developer in talk about design options and client requirements. While the developer was invited to participate in design decisions, he did not invite the designers to participate in implementation decisions. When the UX designers were asked at the time of the observations what the developer was working on, they were not sure. The focus was clearly on the developer being on hand in order to assist with design decisions. Therefore, at this stage of the project the developer and designers engaged on the basis of design decisions that would have implications for implementation.

In Study 1, the developers were doing the work of engaging the UX designers — they moved between floors to find the UX designers and they initiated meetings to get UX design decisions. In this study, **articulation work** was not required for encouraging the UX designers to engage (as it was in Study 1), as much as it was about status updates and clarifications in the interest of workflow. This is closer to the role of the on-going articulation work in Study 2 where the developers and designer shared responsibility for decision-making, and fluidly negotiated dependencies.

6.5 Discussion

As with Study 1 and Study 2, a picture emerges concerning work arrangements, dependencies and mechanisms that make the integration and coordination of work possible. The account in this chapter emphasises the localised, contingent and purposeful work that the developer and designers performed during the two days they were sitting together.

In Study 1, the culture of separation shaped how the developers and designers were going about their work. In Study 2, the developers and designer working closely together shaped their work. In this study, the developer and designers were working within an organisation that although it endorsed separation, was willing to try alternative ways of working and willing to trial Scrum for an entire project. Sitting together shaped how the developer and designers engaged with one another and made progress.

6.5.1 (Re)creating progress

Agile development accepts uncertainty as inherent in software development. Little [2005] notes that uncertainty on Agile development projects can come from various sources: "market uncertainty, technical uncertainty, project duration, and other projects' dependencies on that project." Agile development deals with uncertainty via "iterations, anticipation and adaptation" [Cockburn 2006, p. 259]. The iterative approach allows for feedback and changing client (or Customer) requirements to be incorporated into the software, and the Agile team to adapt prioritisation and estimations of features as appropriate. As far as Agile teams are concerned, the client plays a major part in providing them with certainty in terms of the requirements and in terms of the vision for the software. Indeed, Chow and Cao [2008] have found Customer involvement to be a critical success factor on Agile projects[5].

As mentioned in section 6.4.1, after receiving feedback from the client, Team3

---

[5]The complexities of the Customer role has been investigated by e.g. Martin [2009]'s PhD work.

expressed uncertainty concerning the "vision" for the website and what they were aiming towards. Beck and Fowler [2001, p. 71] note that "There's always some uncertainty within the iteration. Only at the end of an iteration do you really know where you are." However, after the first cycle Team3 were without defined requirements, and without defined tasks for the next cycle. Their client had not provided specificity in terms of what had to be done next. Despite their uncertainty, the developer and designers continued to move forward, avoiding a complete breakdown in the workflow, as a result of the situated work of engaging with one another and creating opportunities for common action.

Making progress was negotiated among the developer and designers such that it was a collective effort, and the meaning of progress was decided among themselves. The developer and designers were collectively (re)creating progress, i.e., between themselves, they projected what was required in the future in terms of what they reconstructed from the past. Engaging with one another and creating opportunities for common action between the developer and designers were mechanisms for responding to the challenging position they found themselves in.

One implication is that the emergent solution is a result of decisions incorporating input from the developer as well as the designers. The progress that was made, was therefore the result of buy-in from both the developer and the designers. Another implication is that the ways in which they were engaging with one another at this stage of the project was setting up subsequent dependencies, and therefore, the subsequent cooperative work relationship between the developer and designers.

6.5.2 Expectations about acceptable behaviour (work group culture)

With respect to how the developers and designers in this organisation interacted outside the trial, the values that shape how developers and designers work together in this organisational setting appears similar to that of the organisation in Study 1. By their own accounts, their "normal" interactions resembled those found in Study 1. The

organisational setting in which Team3 were embedded differed from the organisation of Study 1, in that they displayed a willingness to try something different. As an organisation, they had taken on a project which was going to be different to any other project they had previously been involved in, they were attempting a new project management strategy, and they were bringing the developer and designers into one room. Such decisions could not have been trivial to carry through.

Seating the developer and designers in one room, for example, was accompanied by some organisation-wide upheaval — (1) the room itself required rearranging to accommodate chairs, tables and personal computers for three individuals, and (2) the room had to be booked, which precluded its intended use as a meeting room, or usability testing laboratory. Their efforts can be seen as evidence of organisational values that promote ideas such as learning and adaptability. Further, their efforts indicate a view that places importance on the close interactions between the developers and designers working on a team.

The deliberate decisions in setting up this trial, and specifically the dependencies that resulted from UX taking the lead, shaped expectations about what was acceptable behaviour for the developer and designers. Their assumptions about how they would engage with each other was different from how they "normally" go about their work when sitting apart. Sitting together, the developer and designers seemed to agree that issues would be dealt with as they arose through reciprocated communications, and work updates would be provided as appropriate. From the UX designers' perspective, the developer was to be involved in design decisions, and they expected that the developer would be in a position to give them feedback on the implementation details of the design. From the developer's perspective, the expectation was that the designers would be working through the next version of the visual designs and wireframes, and would ask implementation-related questions as they arose. Team3 followed the option of (re)creating progress in the ways they did, rather than following other options that were open to them, such as doing nothing, or asking the Head of UX for decisions. Their choices are indicative of the culture of this organisation —

the openness, trust, and willingness to respond to each other.

This resembles the interactions between the developers and the designer in Study 2, and contrasts with the expected dependencies between the developers and designers in Study 1, where the designers were in effect independent of the developers — i.e. they were carrying out their design work without consideration for implementation details. In their view, implementation issues could be worked through unproblematically during the implementation cycles. In this study, the designers made their decisions dependent on input from the developer i.e. they chose to check with the developer about what he would find easier to implement, or what he thought would be possible. This supports the assumption that design issues that arise during implementation can potentially be problematic and need to be dealt with during the early stages of the project, and that the developer's input into early design decisions mitigates the risk of serious implementation issues arising later.

6.6 Summary

As with the previous two studies, this study continues in the vein of casting the combination of UX design and Agile development in practice as a problem situated in the organisational setting in which the UX designers and Agile developers are embedded. With organisational support, Team3 were seated together in one room. The team were in an unusual situation in that the client expected the solution to emerge during the project cycles. The team were dealing with uncertainty about what had to happen next, after receiving feedback on their designs from the client. The interactions between the developer and designers and the on-going negotiations of progress allowed the developer and designers to move forward in their work. The team negotiated progress by asking and answering questions, keeping each other informed of decisions, and involving each other in decision-making. In this study, articulation work was instrumental in (re)creating progress despite the uncertainties the team were facing.

This analysis highlighted the notion of shaping subsequent dependencies that would come to bear later in the development effort. The designers were making an effort to design so that the work of implementing their designs would be easy for the developer later on. Also highlighted in this analysis were the expectations about what was acceptable behaviour and how that differed from their normal work, during the time Team3 were sitting together. Team3 involving each other in discussions provides insights into underlying assumptions about each other's availability, ability to answer questions and levels of involvement.

CHAPTER 7

Accounts of practice

7.1 Introduction

This chapter presents the results of the analysis of accounts of practice in the Agile/UX literature. The results in this chapter are distinct from the presentation of the relevant literature that frame the research questions — the literature *review* in chapter 2. This dissertation asks *How are UX design and Agile development combined?* The literature is seen as a source of data for understanding what has been captured about the combination. The research question guiding the analysis was:

> *What are the existing perspectives on combining Agile development and UX design in accounts of practice in the literature?*

which was further broken down into:

- How is the issue of combination conceptualised and how is it addressed?

- What is the experience of using Agile development and UX design together in practice?

The perspective taken in this dissertation, considers combining Agile development and UX design as a challenge that plays out in practice, and that understanding how they are or can be combined requires an understanding of how it is achieved by people, engaging in Agile development and UX design practice. The included accounts from the literature address the research question from this perspective.

The methods set out in this chapter borrow from systematic literature reviews (as described by Kitchenham [2004]), but differ in important ways because we are interested in analysing the papers as accounts of practice rather than aggregating

evidence to answer a specific question. The aim of this analysis is to understand the various ways in which contributors to the literature conceptualise the combination of Agile development and UX design as a challenge and the solutions applied. The research question for this thesis requires looking to the literature to understand what it does answer as much as what it does not answer.

The accounts analysed come from practitioner reports as well as empirical studies. These accounts were analysed separately and presented separately in this chapter, in order to highlight the important differences between them. This analysis shows that while some claims of how Agile development and UX design is combined converge, there remains a messiness around our understanding due to the unresolved conflicts in the accounts. Comparing the practitioner accounts with the accounts in the empirical studies we find that their concerns are similar: getting UX designers and Agile developers to work together requires effort and support in order to achieve their aims. A comparison between practitioner reports and empirical studies presented in section 7.11, highlights the mismatches in the accounts between the two constituencies concerning their understanding of the challenge, and their assumptions about the practical settings in which Agile developers and UX designers work together.

The next sections describe the stages of the method (§ 7.2). Next, the two stages of the search strategy (§ 7.2.1 and § 7.2.2), and the thematic analysis performed on the selected publications (§ 7.2.3) are presented. Sections 7.3 7.6 present the themes that emerged from analysing the practitioner accounts. Sections 7.7 7.9 present the themes that emerged from analysing the empirical study accounts. Section 7.10 discusses the agreements between the practitioner reports and empirical studies. The results from both types of accounts are discussed (§ 7.11), the limitations of the analysis are considered (§ 7.12) and the conclusions presented (§ 7.13).

7.2 Method

This analysis was carried out in distinct stages. The first two stages comprise the search strategy for identifying relevant publications. A set of inclusion and exclusion criteria was applied at each stage of the search, but the final inclusion/exclusion decisions were only made once the full texts had been retrieved during stage 2. The criteria ensured that the selected texts were focused enough to address the research questions. The final stage comprises the detailed analysis.

The analysis of accounts of practice was carried out in three stages as follows:

Stage 1 Retrieve titles and abstracts from electronic databases using queries. Apply inclusion/exclusion criteria.

Stage 2 Apply the quality criteria and determine the type of evidence the account is based on.

Stage 3 Perform a thematic analysis.

Note that although these stages imply that the activities were carried out in a fixed order, there was considerable iteration within and between stages. The next sections describe each stage in detail.

7.2.1 Search strategy: Stage 1

For Stage 1 of the analysis, the following electronic **data sources** were queried:

- ACM Digital Library

- Google Scholar

- IEEE eXplore

- ScienceDirect

- SpringerLink

- Wiley Interscience

- The Open University's electronic library (http://library.open.ac.uk)

Search terms were identified such that the queries with these terms could be used to retrieve the appropriate literature. The terms were taken from the common terms available in the discourse about combining Agile development with *design (see section 1.1 for a discussion on terminology). The initial list of terms was expanded as additional terms were identified during the process of the analysis:

1 agile

2 extreme programming

3 scrum

4 user experience design

5 UX design

6 interaction design

7 user centered design

8 ucd

9 user interface design

10 usability

11 usage centered design

12 user centred design

13 usage centred design

Scrum and XP emerged as important search terms as the search for relevant texts was conducted. One study with Feature-Driven Development appeared in the search results, but was excluded in Stage 2. Other Agile approaches such as Crystal and DSDM did not appear in the search results. To establish whether studies on combining Crystal or DSDM and *design were published, a separate search was conducted. None of the results satisfied the criteria for this analysis. Section 2.5.3 contains a discussion on the availability of studies on DSDM and Crystal.

Using search terms 1–13 above, **queries** were constructed for retrieving publications from the data sources. The following two queries were used to account for the differences in spelling of "centered" and "centred":

query 1: 14 AND 15

where 14 is 1 OR 2 OR 3

and 15 is 4 OR 5 OR 6 OR 7 OR 8 OR 9 OR 10 OR 11

query 2: 16 AND 17

where 16 is 1 OR 2 OR 3

and 17 is 4 OR 5 OR 6 OR 12 OR 8 OR 9 OR 10 OR 13

Based on the output (title and abstract) generated by the queries above, publications were **included** or **excluded** according to the following criteria:

1. include if the publication is in English,

2. include if the full texts are available[1],

3. include if the publication has been peer-reviewed

   - exclude columns, blogs, white papers and prefaces to books.

   - exclude grey literature.

   - exclude publications generated by the research in this dissertation.

The references of the resulting publications were checked for any relevant texts that were not selected by the queries. Stage 1 retrieved a total of 135 texts, of which 30 were eliminated according to the inclusion/exclusion criteria.

---

[1]Availability of a text depended on The Open University's subscription to electronic resources or whether access to paper-based versions could be obtained.

7.2.2 Search strategy: Stage 2

Due to the variety of formats of the abstracts and the type of criteria to be applied, the full texts had to be retrieved in order to evaluate whether the accounts satisfied the quality criteria. The following quality criteria were applied to the 105 publications yielded by stage 1:

- the combination of UX design and Agile development is the main focus of the piece,

- an account of existing or proposed practice is included,

- enough detail to discern how UX designers work or could work with Agile developers is presented.

By applying the above criteria Memmel et al. [2008], for example, is excluded due to the main focus being on tool development, rather than the combination of Agile development and UX design. 'Enough detail' was a qualitative assessment that is difficult to capture. Essentially, the account contained enough details when there was a sense of how UX designers and Agile developers worked together. The account by Chung and Drummond [2009] was excluded on the basis of not having enough detail to discern how the Agile developers and designers worked together. It is clear from the account that the Agile development teams were working with user experience designers. Apart from relating how the user experience designers were having difficulty keeping "the entire user interface design in perspective," or that they were "able to successfully integrate user experience design work into Scrum" there are no details on why the difficulties arose, or how integration was achieved. The quality assessment resulted in another 53 publications being excluded, leaving 52 texts included for the analysis stage.

Stage 2 dealt with publications broadly of two types — practitioner reports and empirical studies — distinguished by the type of evidence on which their accounts of

practice are based. In the case of the practitioner reports, the accounts relate (personal) experience of direct involvement with Agile developers and UX designers in an organisational setting. If practitioner reports satisfied the quality criteria, they were included for analysis in stage 3. Non-practitioner accounts (empirical studies) present accounts that are based on less direct involvement with practice — data collection and analysis as part of a designed research undertaking in the case of the researchers, or the reflection on a combination of theory and techniques/methods/processes applied in a team setting (and not necessarily an organisational setting). All non-practitioner reports were required to relate Agile/UX issues as they were identified based on the data collection and analysis, or the reflection on the outcomes of applying methods/techniques/process. Further analysis was required in order to refine the types of evidence and these details are presented in the section "Different types of evidence".

Grey literature

Previous reviews exclude the *grey literature*, which is a term from Kitchenham [2004] (e.g. Dybå and Dingsøyr [2008]). The central concern in our review is with *accounts of practice* and grey literature was included if the quality criteria were satisfied. For example, the work-in-progress by Lievesley and Yee [2006] has been included in the analysis. The grey literature is indicated in appendix A, and includes a position paper, tutorial descriptions, panel description, work in progress, reports on a tutorial and workshop, demonstration, etc.

Excluded literature

The breakdown of the reasons for excluding literature is given in appendix A. Although the excluded literature was not analysed in this case, these 82 publications play an important role in the Agile development/UX design discourse. The perspective taken in this dissertation, considers combining Agile development and UX design as a problem that plays out in practice, and that understanding how they are/can

be combined requires an understanding of how people achieve this by engaging in Agile development and UX design practice. The criteria presented in this chapter have been designed to select the appropriate accounts from the literature that address the research question from this perspective. However, the Agile/UX literature also contains other perspectives.

Two excluded types of literature with perspectives different to that taken in this dissertation were identified. These are perspectives on the combination of Agile development and UX design that do not require engaging with practice or practical settings, in order to understand how Agile development and UX design can be combined. The first is a group that takes a perspective that relies on practices as they are captured in the literature. These are the accounts by Blomkvist [2005], Memmel et al. [2007b] and Wills et al. [2007], which were counted in the group of nine accounts that were excluded based on having no details of practice in context (§ A.2). The other notable group comprises reviews of the Agile/UX literature. As shown in section A.2 there are two reviews of the literature   the literature review by Sohaib and Khan [2010] and the systematic literature review by da Silva et al. [2011]. Both present a combination of Agile development and UX design based on the findings from their review of the literature. In this dissertation the reviews have been taken into consideration in chapter 2.

Another consideration regarding the excluded literature is that books were excluded from the analysis, in particular a recently published book by Beyer [2010]. However, the analysis includes a paper by Beyer et al. [2004], which takes a similar approach to that found in the book and makes the same salient points.

7.2.3 Thematic analysis: Stage 3

To address the research question for this analysis the focus was both on *what* was written in the accounts and on *how* it was expressed. Common themes that oriented to the research questions were identified across the publications. For example, evidence

emerged from practitioner reports that for Agile developers and UX designers to work together required extra effort. This was particularly evident in publications by Beyer et al. [2004]; Hodgetts [2005]; Miller [2005]; Patton [2002a]; Williams and Ferguson [2007]; Budwig et al. [2009]; Cho [2009]; Federoff and Courage [2009], that contain accounts of initial unhappiness and subsequent adaptations for a better fit [Federoff and Courage 2009], or accounts that make it explicit that with conscious effort they were trying something different to what they had done before [Patton 2002b; Sy 2007]. Together this evidence comes under the theme *Contingency*.

Analysis proceeded iteratively — either confirming or disconfirming through constant comparison between instances. Studies that had generated more than one report, leading to more than one publication included in the analysis, were taken to be evidence of the various ways in which the same problem can be conceptualised and reported on. Therefore, multiple reports of a single study were not discarded.

Considerations for synthesis

When analysing published accounts it is necessary to consider on what basis the elements of those accounts (primary data in the context of this dissertation) are being synthesised. In this analysis the data was synthesised by identifying themes emerging from the accounts of practice in the included publications. Therefore, accounts were not evaluated based on the quality of the study designs reported in the publications, and this was not the basis for the synthesis. Practitioner reports, comprising 29 of the 52 included publications (see table 7.1), can not be considered carefully designed studies. Instead, the descriptive qualities of the accounts were given priority, their similarities and significant differences. This is similar to the approach to synthesising ethnographic accounts by Noblit and Hare [1988] in that the analysis is interpretative rather than aggregative.

Different types of evidence

As mentioned in section 7.2.2, publications were retrieved that contained different sources of evidence on which accounts of practice were based. The two broad categories that were identified in stage 2 (practitioner vs. empirical studies) were refined during the analysis stage. Only one publication fell outside these two categories, that of Parsons et al. [2007], which appeared to be a practitioner report joined on to a study of the literature. This text was excluded from the analysis. The main distinction identified in Stage 2, between practitioner reports and empirical studies, are important to make not only due to the different sources of evidence on which the accounts are based, but also due to the different purposes that these accounts have. While practitioner reports are aimed at an audience of practitioners, the empirical studies are aimed at an audience of practitioners as well as academics.

The thematic analysis preserved the distinction between the different types and then compared the evidence. That is, a thematic analysis was performed with the practitioner accounts and an independent thematic analysis performed with the empirical study accounts. Table 7.1 presents the number of publications included for each type, as well as the corresponding tables providing the reference details of each publication.

Type	Frequency	Details
accounts of practice based on experience (practitioner reports)	29	table 4.2
accounts of practice based on research design (empirical studies)	23	table 4.3
Total	52	

Table 7.1: Accounts of practice.

Practitioner reports

Practitioner reports provide accounts of practice based on (personal) experience of direct involvement with Agile developers and UX designers in an organisational setting. The Agile 2007 conference website calls for practitioner reports and describes what they are and who they are for[2]: "An experience report captures the story of a real agile project, summarizing what happened on the project and the key learning points. These reports allow practitioners to share their practical advice and guidance with other teams." The reports contain clues to the authors' involvement with phrases such as "The different methods we have used..." [Lindström and Malmsten 2008], "... based on our experience..." [Singh 2008] and "... we describe our Project Team's journey" [Frank and Hartel 2009]. In our analysis, they are considered critical, reflective accounts arising from these individuals' participation in software development work, their organisations and communities. Therefore, they represent relevant and valuable evidence about practice. Included in the analysis are 29 practitioner reports shows in table 7.2.

Whether the practitioner reports were written from a UX design perspective or Agile development perspective could not easily be determined. Table 7.2 indicates whether the account is from a UX design perspective. Those texts that are not indicated as from a UX design perspective are not meant to be read as being from an Agile development perspective. Where the perspective is not indicated implies that a definitive decision could not be made based on the information available in the text. Determining that the account was from a UX design perspective required that the author(s) specified role was UX design-related, e.g. *interaction designer, user experience team.*

---

[2]Available at `http://agile2007.agilealliance.org/index.php%3Fpage=session_types%252Fexperience%252Fexperience.html`. Last accessed 22 June, 2011

Empirical studies

Another type of publication encountered in the literature is accounts of practice based on data collection that are part of a deliberate, designed research effort. These accounts arise from data collection, analysis and presentation in the style accepted by the academic community. Included in the analysis are 23 empirical studies. There is a distinction to be made here between two different types of research design. One type of empirical study found in the literature was an exploratory-type study, with the majority conducted in organisational settings. The other type of empirical study were evaluation-type studies, that were practical evaluations of approaches to combining Agile development and UX design, where the approach had been developed outside the setting in which the evaluation took place. Whether the studies are of the exploratory- or evaluation-type is indicated in table 7.3 and the reference for a text used in the dissertation is associated with the reference used in this chapter.

In contrast with the practitioner reports, the empirical study accounts are explicit about the research design (data collection and analysis). Included in the analysis are 12 exploratory-type studies which state how data was collected and analysed: "The authors interviewed three UCD specialists..." [McInerney and Maurer 2005], "We conducted in-depth, semi-structured interviews..." [Fox et al. 2008] and "...using interviews and observations for data collection" [Kollmann et al. 2009]. The exploratory-type empirical studies were qualitative analyses of data from a combination of participant observations and interviews [Chamberlain et al. 2006; Hussain et al. 2009b; Kollmann et al. 2009], interviews [Ferreira et al. 2007a,b; Fox et al. 2008; McInerney and Maurer 2005], participant observations [Haikara 2007] and online surveys [Dayton and Barnum 2009; Hussain et al. 2009c].

The evaluation studies describe how their approaches were designed and then evaluated in practice: "This work presents a development approach that draws from extreme programming (XP), a widely practiced agile software development process, and scenario-based design (SBD), an established usability engineering process" [Lee

and McCrickard 2007]. "We also present the results of two agile projects to validate the proposition that incorporating UCD perspective into Agile Software Development improves the design quality of software systems" [Adikari et al. 2009]. The following combinations appear in the evaluation-type empirical studies: XP and Scenario-Based Design (SBD) [Lee and McCrickard 2007; Lee et al. 2009; Obendorf and Finck 2008], XP and User-Centered Design (UCD) [Hussain et al. 2008b; Narasimhadevara et al. 2008; Wolkerstorfer et al. 2008], XP and Usability Engineering [Memmel et al. 2007c], Scrum and UCD [Paelke and Nebe 2008], Agile and Distributed Cognition [Barksdale et al. 2009], Agile and UCD [Adikari et al. 2009], and Mobile-D and personas [Haikara 2007].

Table 7.2: Accounts of practice from practitioner reports, ordered alphabetically.

Reference for dissertation	Reference for analysis	UX perspective
[Albisetti 2010]	A1	yes
[Armitage 2004]	A2	yes
[Beyer et al. 2004]	A3	yes
[Broschinsky and Baker 2008]	A4´	yes
[Budwig et al. 2009]	A5	yes
[Cho 2009]	A6	yes
[Detweiler 2007]	A7	yes
[Federoff and Courage 2009]	A8	yes
[Fisher and Bankston 2009]	A9	
[Frank and Hartel 2009]	A10	
[Hodgetts 2005]	A11	
[Hussain et al. 2008a]	A12	
[Illmensee and Muff 2009]	A13	yes
[Lievesley and Yee 2006]	A14	yes

Continued on next page

Table 7.2 – continued from previous page

Reference for dissertation	Reference for analysis	UX perspective
[Lindström and Malmsten 2008]	A15	
[Meszaros and Aston 2006]	A16	
[Miller 2005]	A17	yes
[Najafi and Toyoshiba 2008]	A18	yes
[Patton 2002a]	A19	
[Patton 2002b]	A20	
[Patton 2003]	A21	
[Patton 2005]	A22	
[Raithatha 2007]	A23	
[Singh 2008]	A24	
[Sy 2007]	A25	
[Ungar 2008]	A26	yes
[Ungar and White 2008]	A27	yes
[Wilcox et al. 2007]	A28	
[Williams and Ferguson 2007]	A29	yes
Count	29´	

Table 7.3: Accounts of practice from empirical studies, ordered alphabetically.

Reference for dissertation	Reference for analysis	Approach
[Adikari et al. 2009]	A30	evaluation
[Barksdale et al. 2009]	A31	evaluation
[Brown et al. 2008]	A32	exploratory

Table 7.3 – continued from previous page

Reference for dissertation	Reference for analysis	Type of study
[Chamberlain et al. 2006]	A33	exploratory
[Dayton and Barnum 2009]	A34	exploratory
[Ferreira et al. 2007a]	A35	exploratory
[Ferreira et al. 2007b]	A36	exploratory
[Fox et al. 2008]	A37	exploratory
[Haikara 2007]	A38	evaluation
[Hussain et al. 2008b]	A39	evaluation
[Hussain et al. 2009a]	A40	exploratory*
[Hussain et al. 2009b]	A41	exploratory
[Hussain et al. 2009c]	A42	exploratory
[Kollmann et al. 2009]	A43	exploratory
[Lee and McCrickard 2007]	A44	evaluation
[Lee et al. 2009]	A45	evaluation
[McInerney and Maurer 2005]	A46	exploratory
[Memmel et al. 2007a]	A47	exploratory*
[Memmel et al. 2007c]	A48	evaluation
[Narasimhadevara et al. 2008]	A49	evaluation
[Obendorf and Finck 2008]	A50	evaluation
[Paelke and Nebe 2008]	A51	evaluation
[Wolkerstorfer et al. 2008]	A52	evaluation
Count	23	

* Method of analysis not specified

7.3 The challenge represented in practitioner reports

Practitioners experience problems in their organisational setting that in order to address, they combine Agile development with UX design. In the literature this

combination appears as a solution to two problems of software development practice — that of the multiplicity of teams and disciplines involved in software development, and that of improving the product. Their accounts depict their efforts at integrating UX design with Agile development, while at the same time trying to deliver value.

In their accounts, practitioners depict their multidimensional organisational environments as given. That is, software development, as it is practiced in their organisations, requires various teams and various disciplines to combine their efforts in order to deliver software products. For some practitioners, the main aim in their account is to reconcile this multidimensionality in their software development practice. That is, their main aim is to find a way for the various teams and experts to work together [A9], the Agile developers and UX designers representing two of these groups (as we find in the accounts of, for example, Beyer et al. [A3], Hodgetts [A11], Miller [A17]).

Practitioners report on combining Agile development and UX design as a means to deliver a better product for their end users. Alongside of which, practitioners attempt to achieve various other aims, such as improving the impact of the UX designers in the software development process (for example, Sy [A25]), maintaining a "user-orientation" during development [A12], encouraging developers to see the user's perspective [A1], improving communication between the development team and the customer [A4], and improving the working lives of UX designers [A8].

In some sense then, combining Agile development with UX design is itself the problem practitioners are trying to address, as developers and designers working together come to an understanding of how to combine their efforts. In another sense, practitioners are employing the combination of Agile development with UX design to address the basic problem of delivering a better product for their end users. Therefore, Agile development and UX design is one possible solution among several others that could enable teams to deliver a better product. A solution, as we find in the literature, that comes about as a result of organisational decisions into which the teams involved may or may not have had input.

7.4 Integration as focus and coordination

Themes emerging from the practitioner reports show that the specific problems they address in their settings are ones that require addressing notions of *focus* and *coordination*. Practitioners claim that combining Agile development with UX design helps to focus on important aspects of software development. Evident in practitioner accounts is how combining Agile development with UX design requires some notion of ordering, i.e., there is a concern with the way in which Agile development and UX design tasks are organised. Practitioners employ a range of strategies to maintain the appropriate focus and the ordering of tasks in ways that address the various challenges they identify. The strategies are presented in section 7.5.

The central concern of maintaining the appropriate **focus** is to support what is considered the "weaknesses" of one approach with the "strengths" of the other [A3]. Practitioner accounts portray UX design and Agile development as approaches that address different aspects of software development, and as each having a different focus. Agile development is assumed to lack an awareness of usability issues, with little guidance for how and when designers contribute to the process (e.g. [A5]). Practitioner accounts maintain that Agile development does not solve all problems that contribute to software development failing — that Agile development misses some important pieces of the puzzle. According to Patton [A20], XP allowed the team to deliver on time, however, they were still not satisfying the end user. Only by adding Usage-Centered Design could they focus on user goals and focus on what was required to build. The accounts by Miller [A17], Raithatha [A23] and Hussain et al. [A12] similarly acknowledge that focusing on the end user requires that Agile development be combined with UX design.

Besides a focus on end-users practitioner accounts show a variety of ways in which UX design supports Agile development, such as supporting planning and prioritisation [A22], bridging communications between stakeholders [A4] and domains [A27]. Wilcox et al. [A28] also point out the bridging effect of design: "Design, in this and

many more cases, acted as a middle-man between the high-level vision and bottom-up innovations under development."

Maintaining focus is closely linked in practitioner accounts to achieving the **coordination** of UX design and Agile development tasks. For example, from Lindström and Malmsten [A15]'s account, maintaining a focus on end users requires addressing the coordination between developers and designers. Practitioner accounts contain descriptions of sequences of tasks that designers and developers should carry out in order to work together. When bringing Agile development and UX design together, the concern is with allowing the designers and developers to carry out their individual tasks, and then combining their work at the appropriate points. In coordinating sequencing of tasks in a meaningful order practitioners contend with timing and scheduling problems (such as those mentioned in the literature review in chapter 2).

7.5 Combination strategies: Merging, inserting and adapting

Unpicking the different strategies in the literature is not a straightforward task. Although the analysis has identified three main strategies that practitioners use to illustrate how they combined Agile development with UX design, many accounts are combinations of these. There is also evidence in the accounts that the strategies that practitioners present are the result of learning and adapting their approaches (as found in the accounts by Broschinsky and Baker [A4] and Federoff and Courage [A8]). The strategies, as they appear in the accounts, make reference to the processes and practices of Agile development and UX design, and describe merging, inserting and adapting practices from each. Across these strategies, accounts range from precise, step-by-step process descriptions to suggestions for points in the development process where techniques can be inserted.

One strategy is to **merge** the activities of both designers and developers, by specifying the tasks to be performed in sequence, presented as a *process descrip-*

tion. The process descriptions found in the literature arise from the integration of two existing processes, i.e., integrating an established Agile approach (e.g., eXtreme Programming, or Scrum) with an established design approach (e.g., User-Centered Design [A12], Goal-Directed Design [A6] and Contextual Design [A3]).

A second strategy is to **insert** UX design techniques into the process. In this case, the account does not present a merged process description, but contains details of where and when UX design activities were carried out, as an extension of the developers' process. Hussain et al. [A12] claim that XP lends itself to being extended: "Being a lightweight agile process, it is easy to extend the XP process with additional practices." Illmensee and Muff [A13] inserted weekly usability sessions, Cho [A6] included UX designers in Sprint planning meetings where designs were discussed with developers, and Ungar [A26] introduces the design studio in early development cycles. We find examples of established HCI techniques mentioned in the accounts, such as personas [A4] and scenarios [A6]. Singh [A24]'s account inserts a usability role into the Scrum process. Patton [A19] adds the techniques of Usage-Centered Design to the XP process.

The third strategy is to **adapt** UX design techniques. The accounts mention that designers require time at the outset of the project to perform user research and sketch out a coherent design but in order to remain relevant for Agile development, UX design techniques undergo some changes. To fit with the time-boxed Agile cycles, usability techniques are adapted to fit within shorter time-scales and to fit with a "piecemeal" approach to design. Advice is generally to have designers remain ahead of the developers, so that they have enough time to design for what is coming ahead and evaluate what has already been implemented. There are two examples of roles being adapted and redefined. Singh [A24] describes adding a product owner who focuses on usability to a Scrum team, and Williams and Ferguson [A29] divide the UCD role into two roles: one role conducts and communicates the findings of user research studies and usability evaluations, and the other role develops a prototype and maintains close links with the Agile developers.

149

Many accounts are of *UX designers* having to adapt, while it can be inferred from the accounts that adapting is not necessarily only one way. There are examples where developers had to adapt their process in order to have UX designers take part in their planning activities [A10]. Accounts describing changes to the interaction designer role may convey the impression that Agile developers have few or even insignificant adjustments to make when integrating interaction design with Agile development. Underlying all the accounts of practice are the implicit adjustments by Agile developers closer cooperation with interaction designers resulting in less autonomy for the developers than Agile methods suggest and their processes are also becoming heavier up-front (e.g. acknowledgment of Iteration/Sprint Zero as in the account by Sy [A25]).

7.6 Agreement, conflict and contingency

The following collection of themes inform about the experience of combining Agile development with UX design and provides a reflective view on how the solutions played out in practice. The accounts hint at tensions that the combination of Agile development with UX design brings and show how practitioners work through those tensions. While there is agreement among practitioners that combining Agile development with UX design is a positive thing (see section 7.6.1), they disagree about the extent to which combining UX design and Agile development disrupts their software development practice, their organisation, and their teams. There are conflicting perspectives on the implications that the differences and tensions of Agile development and UX design have for practice (see sections 7.6.2 and 7.6.3). Working together holds benefits for practitioners when certain conditions are met; these conditions are discussed in section 7.6.4.

7.6.1 Working together is better and mutually beneficial

According to practitioner accounts, combining Agile development with UX design is something better than what they had been doing before. That practitioner accounts are overwhelmingly in favour of the combination is probably related to the tendency to publish success stories. Not only are their products better, but as a working experience it is presented as preferable to their non-Agile experiences — due to the new opportunities it offers for practitioners to influence the products they are creating. How the experience or the product improves, is not always explained, but practitioners are clear that the combined approach delivers products with higher usability [A24] and better quality of the user experience [A25] than their previous efforts. Meszaros and Aston [A16] claim their usability testing results indicated an improved product. Accounts also refer to increased team confidence [A20], team morale [A10, A12], devotion and satisfaction [A15]. Adapting to working with Agile teams becomes the UCD designers' preferred way of working [A29].

Section 7.4 discussed how Agile development and UX design support each other. Practitioners also report that the combination of Agile development and UX design is mutually reinforcing and achieves intended aims. This is shown in various ways in the accounts: that "The two methods reinforce each other," [A15], working together makes both Agile and UCD stronger and more effective [A27], and that combining Agile and UCD helps keep up with the constantly changing world of web applications [A15].

Benefits to both developers and designers are claimed. In Ungar [A26]'s account the benefits of working together are explained in terms of "role sharing and knowledge transfer." In the account the developers learn the value of UCD and the designers learn about the technical consequences of their designs. In Singh [A24]'s account the developers were more productive, given a clearer vision of usability. Armitage [A2] argues how design is improved with Agile development's ability to learn about changes in requirements sooner in the development effort. In Williams and Ferguson [A29]'s

account, Agile development affords an ability to be responsive to user feedback earlier than on a non-Agile project.

While practitioners present the mutual benefit that combining Agile development with UX design affords, there are accounts that specifically stress the opportunities that the combination brings for UX design [A5, A15, A16, A25]. The organisation's transition to Agile methods brings an opportunity for UX designers to adjust and improve their practices [A25], become better integrated into the culture of their organisation [A16] and become more visible in the organisation [A5].

7.6.2 Natural fit or irreconcilable differences?

That there are benefits to working together is uncontested, however practitioner accounts are not without contradictions. The instances of contradiction from the practitioner accounts show how practitioner viewpoints conflict over whether Agile development and UX design are a natural fit or whether they are irreconcilable approaches. There are contradictions between, and even within, accounts on this theme.

Williams and Ferguson [A29] claim that Agile development and User-Centered Design are a "natural fit." Beyer et al. [A3] claim that Agile development and Contextual Design are also a natural fit. This optimism is supported with claims that approaches which proceed iteratively should work well together. Lindström and Malmsten [A15] state that both Agile and UCD are iterative and that they should work well together. Similarly, Federoff and Courage [A8] refer to the iterative nature of Rapid Iterative Test and Evaluation (RITE) usability testing in support of the claim that "RITE and agile development go hand in hand."

Other practitioner accounts indicate that working together requires designers resigning themselves and adapting their ways of working. Armitage [A2] tells of the frustration designers first felt, before accepting that Agile development requires a different working style. Although Armitage [A2] expresses an opportunity for improving UX design, there is also a sense that the iterative design approach of Agile

development is something to be tolerated. The advice is to adjust UX design practice to suit the nature of Agile development. Federoff and Courage [A8] tell of similar circumstance where the UX team "had no choice but to adapt and evolve."

Contradiction occurring within a single report is that of Williams and Ferguson [A29], who early on in the report claim Agile development and User-Centered Design to be a "natural fit," and then go on to mention the difficulties of the heavy workload of the sole User-Centered Designer on an Agile team, the challenges the team faces "from time to time" and the challenge of the designers trying to stay "at least two iterations ahead of the developers with the design of the UI." The designers' way of working had to be adapted, in their case by dividing the UCD role between two people, bringing into question how "natural" the fit actually is.

Although practitioners may claim that Agile development and UX design fit well together, working together is not free from challenges. Their accounts indicate that significant adaptation is necessary.

7.6.3 Useful tensions or frustrating conflicts?

There are recognised differences between Agile development and UX design. As explained in section 7.4 the different focus of each approach allows practitioners to pay attention to important aspects of software development. Combining Agile development and UX design is intended to enable practitioners to address a broader range of issues, however, their combination is also seen as potentially threatening to the contribution of the other. As Singh [A24] expressed with the first line of their report: "SCRUM poses key challenges for usability." Accounts in the literature show that the differences between Agile development and UX design can act as sources of useful tensions, Lievesley and Yee [A14] claim that the difference in working styles is where the value comes from, while others claim these same differences to be frustrating conflicts to be avoided if possible: "We hope that others can avoid the common pitfalls that we faced in our initial adjustment to agile and scrum" [A5].

One difference between Agile development and UX design cited in the literature is their different approach to design: The iterative, incremental design approach of Agile development as opposed to the holistic, coherent approach of UX design. Armitage [A2] sets out the differences as follows: "Agile methods seek to benefit from the intelligence of experiencing the real product's existence." Whereas "Design, conversely, aims to predict what the entire product will be before it exists." There are positive as well as negative outcomes associated with this difference. In practitioner accounts, Agile teams take advantage of UX design's coherent design approach by allowing UX design to maintain aim/vision for the software [A19, A23, A28, A5, A24, A27, A14].

Maintaining the aim/vision requires coordinating the activities of the UX designers and Agile developers, and so the designers are given a head-start in development. Ungar [A26], for example, gives designers a head-start and allows them to maintain the lead in development, which in turn allows for developing "production-ready software within the time confines of a sprint." The account makes it clear that it can not work any other way. In Budwig et al. [A5]'s account the designers are able to look ahead and anticipate what lies ahead. Patton [A22] shows how Usage-Centered Design supports the planning and prioritisation activities of Agile development.

However, the account by Frank and Hartel [A10] tells of the frustration and problems with team morale that result from excluding developers from the activities that set the aim/vision: "This group began discussing their frustrations with not being included in problem solving while being asked to implement the solution for a feature that only the UXDs, BAs, and/or Product Owner came up with in their preparation for the sprint" [A10]. Another perspective on this difference, which is also presented as a source of frustration, can be found in the account by Federoff and Courage [A8]: "Developers were frustrated that they did not have UI designs that they could start to code in the first sprint and designers felt the one month sprint model did not leave them with enough time to create and evaluate their design." The developers can also become frustrated when the extra work imposed by the designer adds to the pressure

to meet deadlines [A1].

7.6.4 Contingency

Working together holds benefits for practitioners, however, the accounts clearly show this to be the case only if certain conditions are met. Things can go wrong and there are practitioner reports in the literature that acknowledge that. The benefits that are mentioned by Lindström and Malmsten [A15], that is, the "devotion" and "sense of satisfaction" are mentioned alongside the acknowledgment that working together can also be "strenuous" and requires "engagement." Similarly, Armitage [A2] tells of the frustration designers first feel, before accepting that Agile development requires a different working style.

The account by Najafi and Toyoshiba [A18] contrasts the results from two case studies. In the one case, combining UX design with Agile development was a success, whereas in the other case it was not. The success hinged on how UX design was combined with Agile development. In the successful case the User Experience team's participation in the Sprints and Scrums focused the development team on the user experience, whereas on the less successful project, a lack of involvement of the User Experience team led to the project not meeting its targets. Meszaros and Aston [A16] show in their account that UX design can work with Agile development if they are given the chance to carry out their work in the way they are accustomed — that is, with up front research into the domain and with feedback from end-users from usability testing. Therefore, there is a certain way that is preferred and can work, while if this is not adhered to, it will be seen as a failure.

The accounts in the literature provide clues as to what practitioners consider necessary for the combination of Agile development with UX design to work successfully. The discussion in this chapter has already pointed out the various adaptations that were required. While some practitioners consider their fortuitous circumstances as a contributor to their success, other accounts indicate that working together in practice

155

is not achieved spontaneously. The success in the case of Frank and Hartel [A10], was due to strong talented individuals. "Luckily, since we have this existing customer base, we also have a huge pool of users to pull from easily when we need to conduct testing quickly" [A8]. Lievesley and Yee [A14] claim that the flexibility of Agile is better aligned with design thinking, however, the differences in their approaches are "management challenges" that need care and attention to prevent the 'good enough' approach of Agile development from compromising design quality.

In the telling of the ongoing, conscious effort required to work together, practitioner accounts place particular emphasis on notions of *engagement* and *understanding*. The approach taken in the account by Sy [A25] required that designers maintain contact with the developers throughout the development effort. The designers in the account by Broschinsky and Baker [A4] attended developer activities e.g. standups. Lindström and Malmsten [A15] require that there is "no hiding", "engagement from the whole team", "management support and trust" and "an understanding of one's colleagues and their different competences." For Beyer et al. [A3], Agile developers and UX designers working together successfully requires an understanding of the underlying assumptions and experiences of both approaches.

Practitioner accounts show that there is also some persuading work to do on the part of the UX designers. Albisetti [A1] tell how the designers had to justify their decisions to the developers. The account by Lievesley and Yee [A14] tells how the designers had to overcome the skepticism among the developers, and establish the value they bring to the project. There is evidence that developers gradually realise the value of design work (for example in the account by Ungar and White [A27]). How designers communicate findings from usability investigations to developers is also mentioned as important [A27, A25]. Broschinsky and Baker [A4] sum up an attitude prevalent in the practitioner accounts: Although a Human Factors specialist is seen as "someone who adds work" they are undeniably a key ingredient to success — even "if they [development managers] as yet don't understand all those magic ingredients."

The preceding discussion shows that, given the differences between Agile development and UX design, harnessing these differences in order to achieve a beneficial outcome requires careful, *local* management and attention.

7.7 The challenge represented in empirical studies

Two types of empirical studies were found in the literature: exploratory studies of Agile development and UX design practice, and empirical evaluations of methods or processes. Both types of studies formulate their research questions independently of practical settings. The questions they address are not drawn from the practical settings that they study, but from other sources, such as other empirical studies [A30], debates between experts [A35, A36], experience [A34], or a synthesis of existing knowledge about practice [A31]. One exception is the study by Chamberlain et al. [A33], which identifies problems via observations of practice and offers advice for those problems.

All of the empirical studies take as their subject of investigation the *differences* between Agile development and UX design with an emphasis on overcoming or reconciling those differences. In the accounts, the differences are expressed as the tensions that arise from the different perspectives of Agile development and UX design, or expressed as Agile development "lacking" perspectives on users and usability (as expressed in the accounts by Haikara [A38], Memmel et al. [A48] and Obendorf and Finck [A50], for example). While some empirical studies aim for a better understanding of those differences, others propose solutions in the form of integrated Agile development and UX design processes. For example, Brown et al.[A32] set out to understand collaboration between Agile developers and UX designers, while the studies by Lee and McCrickard [A44] and Hussain et al. [A39] propose process descriptions that claim to reconcile goals, motivations and perspectives. A selection of empirical studies present an appraisal of the similarities and differences between Agile development and UX design (as found in the accounts by Chamberlain et al. [A33], McInerney

and Maurer [A46] and Memmel et al. [A47], for example). These appraisals are analyses of the accepted knowledge and expertise in the literature (such as that of Beck [1999] and Constantine [2002]). This appraisal then serves as motivation for carrying out the studies.

Combining Agile development with UX design is itself the problem under investigation for one group of empirical studies. These studies contain accounts of achieving "integration" between Agile development and UX design (as found in the studies by Fox et al. [A37] and Memmel et al. [A47]). For another group the problem of integration resides in some larger overarching research context. The overarching contexts that appeared in the accounts were:

- Bridging the gap between software engineering and HCI [A33, A41, A37, A47],

- understanding and facilitating collaboration between developers and designers [A31, A32],

- understanding and valuing usability professionals [A34, A43],

- designing systems, such as mixed-reality systems [A51] and support for transplant nursing [A49].

7.8 Understanding differences and similarities between Agile development and UX design

There is considerable variation in how empirical studies address the combination of Agile development and UX design, each with its own perspective on what aspects of the combination require investigation and emphasis. However, a common assumption among the empirical studies is that an understanding of how Agile development and UX design can be combined, requires an understanding of what is similar or different between them.

As mentioned in section 7.7, the empirical studies are motivated by the differences

between Agile development and UX design. In the accounts, assumptions about Agile development and UX design, understood as free from any context (i.e., without reference to practical settings) are compared, and based on these comparisons considered more or less ideal for integration.

The iterative nature of UX design and Agile development is a commonly cited similarity in the empirical study accounts. According to Hussain et al. [A41], their iterative nature is "key" to the approaches integrating in an unproblematic way. McInerney and Maurer [A46] observe that the artefacts that Agile development and UCD iterate on are different: the UCD approach evaluates "low-fi designs" versus Agile development's evaluation of "production-ready code." Ferreira et al. [A35] take this observation further, to say that the nature of what is iterated *on* could impact on the synchronisation between Agile development iterations and UX design iterations: "while agile methods iterate on code with iterations lasting weeks, UI designs typically iterate only on the user interface using low technology prototypes with iterations lasting hours or days." There is little empirical evidence in the accounts of iterations either matching or being out of synch and what the implications are for practice. Based on what little empirical evidence exists the coordination between developers and designers is presented as an unproblematic back-and-forth between design and implementation throughout the development effort.

UX design and Agile development are claimed to differ based on the amount of work that is acceptable to carry out before coding begins. According to Chamberlain et al. [A33], the UCD team requires time before the developers start coding, to "understand their users as much as possible," while Agile development views this as time spent "at the expense of writing code." Lee et al. [A45] observe that this difference is a consequence of the values of Agile development: "working software over comprehensive documentation and responding to change over following a plan." Ferreira et al. [A36] trace this particular conflict about what should occur "up front" in an Agile development project to the well-known debate between Kent Beck and Alan Cooper [Nelson 2002], who disagreed. Empirical studies confirm that up-front

work is carried out when Agile development is combined with UX design in practice (as found by Fox et al. [A37], Hussain et al. [A41] and Ferreira et al. [A36]). Apart from the empirical evidence for up-front work, there is little discussion about what the discrepancy between out of context comparisons between Agile development and UX design implies for an understanding of practice.

McInerney and Maurer [A46] claim that the differences in the literature are not reflected in practice: "Examining professional practice, as we do in this article, paints a different picture of how UCD and agile practices coexist in a development team." No other empirical study accounts relate their empirical findings back to the out of context comparisons between Agile development and UX design. The account by Chamberlain et al. [A33] identify challenges for the combination of UCD and Agile development from the practical setting under study, however, they are silent on how these relate to the similarities and differences that had been identified out of context. The literature is silent on the implications of comparing Agile development and UX design out of context, and what it may mean for empirical evidence of practice.

7.9 (Un)problematic combinations

Empirical study accounts give the impression that it is possible to slot the activities of the UX designers and Agile developers together in a way that suggests it is unproblematic to do so [A42]. The empirical studies suggest that the solution to integrating UX design and Agile development is to find the right process. As Lee et al. [A45] reconcile the "competing aims" of Agile developers and UX designers with a process called "eXtreme Scenario-based Design." Hussain et al. [A40] claim: "We found no cultural difference [between usability person and developers]," which supports an unproblematic view on Agile developers and UX designers working together.

However, not all empirical studies agree with this view. There is also evidence that views on what generates value in software development practice complicates attempts at integration. The account by Dayton and Barnum [A34] illustrates this imbalance:

"Unless a team member argues successfully for making usability testing a priority, it is unlikely to be included because the always urgent, overriding goal is developing working features at the end of each sprint cycle. Agile methodologies have thus changed the development process in ways that provide difficult new challenges for user experience practitioners." Kollmann et al. [A43] present a similar finding: "delivery has priority, so user research is the first thing to be compromised." Haikara [A38] claims that adding personas to the Agile development effort was a success, while acknowledging that the developers needed constant reminding and encouragement to make use of the personas in their work.

McInerney and Maurer [A46] point to the lack of support in the literature as the reason the UCD roles have difficulties finding their place in Agile development : "Like other development methods, the agile literature does not identify a distinct UCD role, so the onus remains on UCD to justify and define its role on the team." While one set of empirical study accounts suggest integrating Agile development and UX design relies on finding a process that has the right kind of UX design and Agile development tasks, performed in the right sequence, another set of empirical study accounts hint at the challenges that Agile developers and UX designers face — challenges that arise from Agile development and UX design not always being on equal footing in practical settings.

7.10 Shared concerns between practitioner reports and empirical studies

Practitioner accounts and empirical study accounts exhibit evidence of common concerns — they agree that there are issues around the timing and scheduling of tasks and focus on solutions in the form of processes, methods and tools. The issues around the timing and scheduling of the tasks of Agile developers and UX designers have been found in the practitioner as well as empirical study accounts to be non-trivial. The practitioner accounts and empirical study accounts agree that combining Ag-

ile development with UX design brings problems for involving users [A7, A52] and incorporating their feedback back into the development effort [A7, A33, A44, A46, A52]. They agree on problems with coordinating between the activities of the UX designers, Agile developers and other non-Agile teams [A5, A7, A37], and performing usability testing in the context of other Agile development tests, e.g. unit testing and acceptance testing [A18, A44, A52].

Practitioner accounts and empirical study accounts overlap in how they present solutions to address the various challenges of practice that are reported in the accounts. While the details vary, these solutions take the form of processes, methods and tools. Section 7.5 discussed the strategies that practitioner accounts employ to illustrate how Agile development is combined with UX design. The merging strategy relies on processes, the inserting strategy relies on discrete tools and methods, while the adapting strategy adapts discrete tools and methods. The evaluation-type empirical studies all focus on a process (e.g. eXtreme Scenario-based Design [A45]) or a tool (e.g. personas [A38]) to evaluate in practice. The exploratory-type empirical studies focus less on processes, tools and methods but may contain recommendations for processes, tools and methods, e.g. "To communicate the UX vision, personas and scenario-based approaches have been successful," [A43], "Schedule usability testing, regularly, and routinely," [A34].

7.11 A disjointed discourse

The separation between practitioner accounts and empirical studies has been preserved in this analysis, due to their different sources of evidence and different purposes. Section 7.7 has shown that there is also a difference in their *motivations* for investigating Agile development and UX design, and the overarching context in which the problem resides. In the practitioner accounts the agreements, conflicts and contingencies in section 7.6 all stemmed from the practitioners' aim to maintain focus and achieve coordination between the developers and designers. The emphasis in the

empirical studies is on the difference between Agile development and UX design.

Comparing the practitioner accounts with the accounts in the empirical studies we find that their concerns are similar: getting UX designers and Agile developers to work together requires effort and support in order to achieve their aims. The practitioner and empirical study accounts agree that addressing the problems that combining Agile development with UX design brings, relies on finding the right process, tools and methods. Practitioner accounts present processes, tools and methods for maintaining focus and coordinating between tasks when integrating Agile development with UX design, while accounts in the empirical studies present processes, tools and methods for overcoming the differences between Agile development and UX design. However, the ways in which the accounts diverge highlight the disjointedness in the discourse on Agile development and UX design practice. This disjointedness is particularly evident in how the accounts treat the differences between Agile development and UX design, and the assumptions about the status of UX design and Agile development in organisations.

Compared to accounts in empirical studies, practitioner reports place less emphasis on trying to understand the differences between Agile development and UX design, and are less concerned with how other reports construct the similarities and differences between them. Practitioner accounts emphasise their immediate experiences of solving problems with individuals who contribute in different ways to development of the software. There are few practitioner reports containing similar appraisals, although they may also refer to experts — such as Constantine and Lockwood [1999] referenced in the report by Patton [A21] (currently considered an expert himself), and Beck [1999] referenced in the reports by Hussain et al. [A39] and Armitage [A2]. This suggests a tension between presenting an understanding of the differences between Agile development and UX design as it is read in the literature, and tensions that manifest in practical settings.

Comparing practitioner accounts with accounts in the empirical studies, shows how assumptions about practice can be glossed over. This is especially true with

respect to the issue of up-front design. In practitioner accounts, up-front design appears as one of the conditions for success and helps to maintain aim/vision. There is no discussion about whether it should or should not be done. Practitioners ask how designers fit into the Agile process. It can also be framed in terms of a power struggle. That UX designers often have to persuade their organisation, convince developers that they are relevant and need enough time to do their work. Empirical studies design processes that assume a combination based on development and design being on an equal footing. That UX design and Agile development may not enjoy an equal balance of power in an organisational setting has been identified in the study by Kollmann et al. [A43]. Aside from that study, there is little evidence to be found in other empirical studies that takes this struggle for relevance into account.

Clearly, practical settings as portrayed in practitioner accounts and assumptions about those settings in empirical studies, are not aligned. Given that only four empirical studies included in this analysis, included observations of practice, the apparent lack of understanding of what goes on in practice comes as no surprise. The themes emerging from the analysis in this chapter do not explain what shapes the combination of Agile development and UX design in practice. Neither do they explain how Agile development and UX design work is accomplished on a day-to-day basis.

7.12 Limitations

The following limitations of the analysis of accounts of practice should be taken into consideration:

1. The literature selected according to the approach described in this chapter may suffer from *publication bias* [Kitchenham 2004]. That is, the majority of the texts included in the analysis can be seen as "success stories." However, included in the analysis are accounts that report on the challenges faced (e.g. frustrations) and how these were overcome.

2. The texts are all in electronic form.

3. Non-English language texts were excluded, e.g., the account by Holzinger and Slany [2006] appeared in the stage 1 search results, but was excluded due to it being in German. The researcher discussed these with an individual whose first language was German to ensure that significant findings were not missed.

4. The queries resulted in a majority of texts using XP and Scrum. Queries with approaches other than XP and Scrum were run during the analysis to ensure significant accounts were not missed, however, these accounts did not satisfy the criteria for inclusion in the analysis. The nature of the Agile/UX literature is that XP and Scrum dominate empirical studies and practitioner reports.

5. This research deals with terms that are not well-defined or standardised in the literature. Running the queries with the search terms selected for this analysis, therefore, may have omitted relevant studies. Checking the references of the included texts for other relevant texts helped to ensure significant accounts were not missed.

7.13   Conclusion

Within each group of accounts — the practitioner accounts and empirical study accounts — certain claims about combining Agile development with UX design converge (e.g. the combination strategies in the practitioner accounts and the treatment of the differences between Agile development and UX design in the empirical study accounts). There also is a coherence within each group concerning the type of problem each considers the combination to be. However, unresolved conflicts remain within and between these groups. This analysis has highlighted where the discourse regarding practice becomes disjointed: in the treatment of the differences between Agile development and UX design, and the assumptions about the status of UX design and Agile development in organisations.

Based on the findings of this analysis, it is clear that more work is needed to understand the settings in which Agile development is combined with UX design, and how the setting shapes practice. Further research into combining Agile development with UX design, therefore, depends on researchers not only engaging with practitioners, but specifically engaging with the settings in which practitioners work.

7.14 Accounts included in the analysis

Practitioner reports

A1   M. Albisetti. Launchpad's quest for a better and agile user interface. In W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, A. Sillitti, A. Martin, X. Wang, and E. Whitworth, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 48 of Lecture Notes in Business Information Processing, pages 244–250. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13054-0. URL http://dx.doi.org/10.1007/978-3-642-13054-0_26.

A2   J. Armitage. Are Agile methods good for design? *interactions*, 11(1):14–23, 2004. ISSN 1072-5520. doi: http://doi.acm.org/10.1145/962342.962352.

A3   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 '04: Extreme Programming and Agile Methods, Calgary, Alberta, Canada, August 1518*, pages 50–59. vol. 3134 of Lecture Notes in Computer Science, Springer, 2004.

A4   D. Broschinsky and L. Baker. Using persona with XP at LANDesk Software, an Avocent company. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 543–548, Los Alamitos, CA, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3321-6. doi: http://doi.ieeecomputersociety.org/10.1109/{A}gile.2008.91.

A5   M. Budwig, S. Jeong, and K. Kelkar. When user experience met agile: a case study. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI '09, pages 3075–3084, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4. doi: http://doi.acm.org/10.1145/1520340.1520434.

A6   L. Cho. Adopting an agile culture. *AGILE Conference*, 0:416–421, 2009. doi: http://doi.ieeecomputersociety.org/10.1109/AGILE.2009.76.

A7   M. Detweiler. Managing UCD within Agile projects. *interactions*, 14(3):40–42, 2007. ISSN 1072-5520. doi: http://doi.acm.org/10.1145/1242421.1242447.

A8   M. Federoff and C. Courage. Successful user experience in an agile enterprise environment. In *Proceedings of the Symposium on Human Interface 2009 on Conference Universal Access in Human-Computer Interaction. Part I: Held as Part of HCI International 2009*, pages 233–242, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02555-6. doi: http://dx.doi.org/10.1007/978-3-642-02556-3_27.

A9   K. Fisher and A. Bankston. From cradle to sprint: Creating a full-lifecycle request pipeline at nationwide insurance. In *Agile Conference, 2009. AGILE '09.*, pages 223–228, Aug 2009. doi: 10.1109/AGILE.2009.72.

A10   A. Frank and C. Hartel. Feature teams collaboratively building products from ready to done. In *Agile Conference, 2009. AGILE '09.*, pages 320–325, Aug 2009. doi: 10.1109/AGILE.2009.51.

A11   P. Hodgetts. Experiences integrating sophisticated user experience design practices into Agile processes. In *Agile '05: Proceedings of the 2005 Agile Conference, Denver, CO, USA. July 24–29*, pages 235–242, Los Alamitos, CA, USA, 2005. IEEE Computer Society. doi: 10.1109/ADC.2005.24.

A12   Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, and P. Wolkerstorfer. Agile user-centered design applied to a mobile multimedia streaming application. In A. Holzinger, editor, HCI and Usability for Education and Work, volume 5298 of Lecture Notes in Computer Science, pages 313–330. Springer Berlin / Heidelberg, 2008a. ISBN 978-3-540-89349-3. URL http://dx.doi.org/10.1007/978-3-540-89350-9_22.10.

A13   T. Illmensee and A. Muff. 5 users every friday: A case study in applied research. In *Agile Conference, 2009. AGILE '09*, pages 404–409, 2009. doi: http://10.1109/AGILE.2009.45.

A14   M. A. Lievesley and J. S. R. Yee. The role of the interaction designer in an Agile software development process. In *CHI '06: CHI 06 extended abstracts on Human factors in computing systems*, pages 1025–1030, New York, NY, USA, 2006. ACM. ISBN 1-59593-298-4. doi: http://doi.acm.org/10.1145/1125451.1125647.

A15   H. Lindström and M. Malmsten. User-centred design and agile development: Rebuilding of the Swedish national union catalogue. *The Code4Lib Journal*, 5:1215, 2008. ISSN 1940-5758.

A16   G. Meszaros and J. Aston. Adding usability testing to an agile project. In *Proceedings of the conference on AGILE 2006*, pages 289–294, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2562-8. doi: `10. 1109/AGILE.2006.5`.

A17   L. Miller. Case study of customer input for a successful product. In *Agile '05: Proceedings of the 2005 Agile Conference, Denver, CO, USA, July 24– 29*, pages 225–234, Los Alamitos, CA, USA, 2005. IEEE Computer Society. doi: `10.1109/ADC.2005.16`.

A18   M. Najafi and L. Toyoshiba. Two case studies of user experience design and Agile development. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada*, August 4–8, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

A19   J. Patton. Hitting the target: Adding interaction design to Agile software development. In *OOPSLA '02: OOPSLA 2002 Practitioners Reports*, pages 1–7, New York, NY, USA, 2002a. ACM. ISBN 1-58113-471-1. doi: `http: //doi.acm.org/10.1145/604251.604255`

A20   J. Patton. Designing requirements: Incorporating usage-centered design into an agile sw development process. In D. Wells and L. Williams, editors, *Extreme Programming and Agile Methods. XP/Agile Universe 2002, volume 2418 of Lecture Notes in Computer Science*, pages 95–102. Springer Berlin / Heidelberg, 2002b. doi: `http://dx.doi.org/10.1007/3-540-45672-4_1`.

A21   J. Patton. Improving on agility: Adding usage-centered design to a typical Agile software development environment. In *ForUse2003: Proceedings of the Second International Conference on Usage-Centered Design*, Portsmouth, NH, USA, October 18–22, 2003.

A22   J. Patton. Finding the forest in the trees. In *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, OOPSLA '05, pages 266– 274, New York, NY, USA, 2005. ACM. ISBN 1-59593-193-7. doi: urlhttp://doi.acm.org/10.1145/1094855.1094961.

A23   D. Raithatha. Making the whole product agile: a product owners perspective. In *Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming*, XP'07, pages 184–187, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-73100-9. http://portal.acm.org/citation.cfm?id=1768961.1769003.

A24   M. Singh. U-SCRUM: An Agile methodology for promoting usability. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 555–560, Los Alamitos, CA, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3321-6. doi: http://doi.ieeecomputersociety.org/10.1109/{A}gile.2008.33.

A25   D. Sy. Adapting usability investigations for Agile user-centered design. *Journal of Usability Studies*, 2(3):112–132, May 2007.

A26   J. Ungar. The design studio: Interface design for Agile teams. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 519–524, Los Alamitos, CA, USA, 2008. IEEE Computer Society. doi: 10.1109/{A}gile.2008.51.

A27   J. Ungar and J. White. Agile user centered design: Enter the design studio — a case study. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2167–2178, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358650.

A28   E. Wilcox, S. Nusser, J. Schoudt, J. Cerruti, and H. Badenes. Agile development meets strategic design in the enterprise. In G. Concas, E. Damiani, M. Scotto, and G. Succi, editors, *Agile Processes in Software Engineering and Extreme Programming*, Lecture Notes in Computer Science, pages 208–212. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2007.

A29   H. Williams and A. Ferguson. The ucd perspective: Before and after agile. In *AGILE 2007*, pages 285–290, Aug 2007. doi: 10.1109/AGILE.2007.61.

Empirical studies

A30 S. Adikari, C. McDonald, and J. Campbell. Little design up-front: A design science approach to integrating usability into agile requirements engineering. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part I: New Trends*, pages 549–558, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02573-0. doi: http://dx.doi.org/10.1007/978-3-642-02574-7_62.

A31 J. Barksdale, E. Ragan, and D. McCrickard. Easing team politics in agile usability: A concept mapping approach. In *Agile Conference, 2009. AGILE '09*, pages 19–25, Aug 2009. doi: 10.1109/AGILE.2009.57.

A32 J. Brown, G. Lindgaard, and R. Biddle. Stories, sketches, and lists: Developers and interaction designers interacting through artefacts. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4-8*, pages 39–50, Los Alamitos, CA, USA, 2008. IEEE Computer Society. doi: 10.1109/{A}gile.2008.54.

A33 S. Chamberlain, H. Sharp, and N. A. M. Maiden. Towards a framework for integrating Agile development and user-centred design. In *XP '06: Proceedings of the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering, Oulu, Finland, June 17-22*, pages 143–153, 2006.

A34 D. Dayton and C. Barnum. The impact of agile on user-centered design: Two surveys tell the story. *Technical Communication*, 53(3):219–234, August 2009.

A35 J. Ferreira, J. Noble, and R. Biddle. Agile development iterations and ui design. In *Agile '07: Proceedings of the Agile 2007 Conference, Washington, DC, USA, August 13-17*, pages 50–58, Los Alamitos, CA, USA, 2007a. IEEE Computer Society. ISBN 0-7695-2872-4. doi: http://dx.doi.org/10.1109/AGILE.2007.8.

A36 J. Ferreira, J. Noble, and R. Biddle. Up-front interaction design in Agile development. In G. Concas, E. Damiani, M. Scotto, and G. Succi, editors, *Agile Processes in Software Engineering and Extreme Programming*, pages 9–16. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2007b. ISBN 978-3-540-73100-9. doi:10.1007/978-3-540-73101-6_2.

A37 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 '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4-8*, pages 63–72, Los Alamitos, CA, USA, 2008. IEEE Computer Society. doi: 10.1109/Agile.2008.54.

A38   J. Haikara. Usability in Agile software development: Extending the interaction design process with personas approach. In G. Concas, E. Damiani, M. Scotto, and G. Succi, editors, *Agile Processes in Software Engineering and Extreme Programming*, Lecture Notes in Computer Science, pages 153–156. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2007.

A39   Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, and P. Wolkerstorfer. Integrating Extreme Programming and User-Centered Design. In *PPIG '08: Proceedings of the 20th annual meeting of the Psychology of Programming Interest Group, Lancaster, UK*, pages 107–113. Lancaster University, 2008b.

A40   Z. Hussain, H. Milchrahm, S. Shahzad, W. Slany, M. Tscheligi, and P. Wolkerstorfer. Integration of extreme programming and user-centered design: Lessons learned. In *Agile Processes in Software Engineering and Extreme Programming*, Lecture Notes in Business Information Processing, pages 174–179, 2009a. doi: 10.1007/978-3-642-01853-4_23.

A41   Z. Hussain, W. Slany, and A. Holzinger. Investigating agile user-centered design in practice: A grounded theory perspective. In *Proceedings of the 5th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for e-Inclusion*, USAB '09, pages 279–289, Berlin, Heidelberg, 2009b. Springer-Verlag. ISBN 978-3-642-10307-0. doi: http://dx.doi.org/10.1007/978-3-642-10308-7_19.

A42   Z. Hussain, W. Slany, and A. Holzinger. Current state of agile user-centered design: A survey. In *Proceedings of the 5th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for e-Inclusion*, USAB '09, pages 416–427, Berlin, Heidelberg, 2009c. Springer-Verlag. ISBN 978-3-642-10307-0. doi: http://dx.doi.org/10.1007/978-3-642-10308-7_30.

A43   J. Kollmann, H. Sharp, and A. Blandford. The importance of identity and vision to user experience designers on agile projects. In *Proceedings of the 2009 AGILE Conference, Chicago, IL, USA, August 24–28*, pages 11–18. IEEE Computer Society, 2009. ISBN 978-0-7695-3768-9. doi: http://doi.ieeecomputersociety.org/10.1109/AGILE.2009.58.

A44   J. C. Lee and S. McCrickard. Towards extreme(ly) usable software: Exploring tensions between usability and Agile software development. In *Agile '07: Proceedings of the Agile 2007 Conference, Washington, DC, USA, August 13–17*, pages 59–71, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

A45   J. Lee, D. Scott McCrickard, and K. Stevens. Examining the foundations of agile usability with extreme scenario-based design. In *Agile Conference, 2009. AGILE '09*, pages 3–10, Aug 2009. doi: 10.1109/AGILE.2009.30.

A46   P. McInerney and F. Maurer. UCD in Agile projects: Dream team or odd couple? *interactions*, 12(6):19–23, 2005.

A47   T. Memmel, F. Gundelsweiler, and H. Reiterer. Agile human-centered software engineering. In *HCI '07: Proceedings of 21st BCS HCI Group conference, University of Lancaster, UK, September*, pages 167–175. British Computer Society, 2007a.

A48   T. Memmel, H. Reiterer, and A. Holzinger. Agile methods and visual specification in software development: a chance to ensure universal access. In *Proceedings of the 4th international conference on Universal access in human computer interaction: coping with diversity*, UAHCI07, pages 453–462, Berlin, Heidelberg, 2007c. Springer-Verlag. ISBN 978-3-540-73278-5. URL http://portal.acm.org/citation.cfm?id=1766311.1766364.

A49   A. Narasimhadevara, T. Radhakrishnan, B. Leung, and R. Jayakumar. On designing a usable interactive system to support transplant nursing. *J. of Biomedical Informatics*, 41:137–151, February 2008. ISSN 1532-0464. doi: 10.1016/j.jbi.2007.03.006.

A50   H. Obendorf and M. Finck. Scenario-based usability engineering techniques in Agile development processes. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2159–2166, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358649.

A51   V. Paelke and K. Nebe. Integrating Agile methods for mixed reality design space exploration. In *DIS '08: Proceedings of the 7th ACM conference on Designing interactive systems*, pages 240–249, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-002-9. doi: http://doi.acm.org/10.1145/1394445.1394471.

A52   P. Wolkerstorfer, M. Tscheligi, R. Sefelin, H. Milchrahm, Z. Hussain, M. Lechner, and S. Shahzad. Probing an Agile usability process. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2151–2158, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358648.

CHAPTER 8

Discussion

8.1 Introduction

In this chapter the findings from the field work and the analysis of the accounts of practice selected from the literature are discussed in terms of how they answer the overarching research question: *How is Agile development combined with UX design?* There are important distinctions to be drawn between how accounts in the literature address this question and the empirical evidence from the field work. In shaping practice, the studies that were conducted in organisational settings show that setting has a much more significant role than accounts of practice suggests. Integrating Agile development and UX design is negotiated and achieved on a day-to-day basis between the developers and designers in those settings, such that integration is an ongoing achievement in practice.

This is not an investigation into whether Agile developers and UX designers should be co-located or not. The UX designers who were working closely with the Agile developers were observed in settings where the designers were seated on a different floor to the developers and where the designers and developer were working in one room. The togetherness or separation between the developers and designers (as will be discussed in this chapter) depends on contextual values endorsed by the organisation, regardless of whether the designers and developers were sitting together or not. Agile developers and UX designers do what is required in order to get their job done, by maintaining focus and coordinating between their tasks, expecting certain behaviours from others, maintaining mutual awareness, negotiating progress and engaging with each other. Their work in each setting depends on the values endorsed by the organisations in which the developers and designers are embedded.

This chapter is structured as follows: Based on the analysis of the field work, the decisions and underlying values and assumptions shaping the work of the developers and designers are presented (§ 8.2). The next section links the understanding of Agile/UX integration gained from practical settings to issues identified in the analysis of accounts of practice (§ 8.3). Next, the major themes that emerged from the analysis

of the field work and the analysis of the accounts of practice are presented to show how integration is an on-going achievement in practice (§ 8.4). Section 8.5 considers the limitations for generalising from this research and the key findings are summarised (§ 8.6).

8.2 Shaped by decisions

The purpose of this section is two fold. First, the discussion extends the thematic analysis of the studies of Team1, Team2A, Team2B and Team3 by explicating how values endorsed by the teams' settings shaped their work, workflow, and ultimately, Agile development and UX design practice. Second, the discussion explains the "why" aspect of the overarching research question and, specifically, the research subquestion that asks: *What shapes the combination of Agile development and UX design in practice?*

This discussion assumes that underlying the decisions concerning how Agile development and UX design should work together in the organisation, are values and assumptions that motivate those decisions — a value being a belief about what ought to be [Schein 2010, p. 25]. This relationship between decisions, values and assumptions has been studied by other researchers as *levels of organisational culture*, the visible aspects of which are observable in its *artefacts* [Schein 1990]. The research in this dissertation was not aimed at understanding organisational culture and yet elements that can be associated with culture showed up in the analysis of the data and were significant in explaining how and why the Agile developers and UX designers in each team were working together in the ways we had observed. In the rest of this section we trace decisions that we learned about during our field work to values and assumptions underlying the decision-making and maintained through the work by the developers and designers. These values and assumptions relate to combining Agile development with UX design.

Much of our understanding about the work practices of the teams we observed in

Studies 1, 2 and 3 was informed by the visible activities that we as observers were given access to. For example, in Study 1 the developers were sitting apart from the designers and we observed the developers walking around and going up the stairs. In Study 2, a similar seating arrangement meant there were similar activities to observe. Whereas in Study 3, there was no walking around due to the designers and the developer sitting in the same room. Through the ongoing discussions with the participants it became obvious how these visible activities were the consequences of decisions by management roles *external* to the teams. The consequences of decisions that resulted in the developers and designers sitting together extended not only to whether they were walking around or not, but the decisions, and the values and assumptions underlying them, held far broader consequences for how the work that was observed could be understood as the integration of Agile development and UX design. Everything that had been observed was no mere accident, and was not solely determined by the participants' understanding of their development "process". What the teams were doing in practice can be accounted for to a significant degree by views on software development that did not arise from within the team, or from the day-to-day business of software development.

8.2.1 Study 1: Valuing separation

For Team1, transforming the UX designs into functional software occurred as distinct phases of activity. In the first phase the developers spent significant time becoming familiar with the UX designs and building up an understanding of what the designs meant in terms of working software. The second phase of activity involved inspecting the designs handed over by the UX designers, comparing it to the software already implemented and identifying mismatches between them. In the final phase the implementation work had to be added to the existing Scrum backlog.

A member of Team1 informed us that the decision to separate UX designers and developers was put into effect by management in the UX division. This was based

on their view that UX designers work best when they are separated from the issues of software construction, which hamper their creativity. In light of this decision and our observations of practice, we can explain management's view on how to achieve quality software, as well as the place of UX design relative to software construction, in terms of valuing the separation between the Agile and UX disciplines. This was at odds with the view held by the Agile developers in the organisation, but endorsed by the UX designers.

In this view, UX design exists independently from construction. Optimal UX design is created in a "design phase" in which UX designers are free to apply their creative energies without considerations about whether the designs can be turned into functional software. Agile development applies to the "construction phase" during which UX designs are implemented as functional software. Agile development becomes the mechanism through which UX designs are turned into functional software. This view has nothing to say about UX design issues emerging from software construction and, consequently, it has nothing to say about the notion of emerging requirements. The expectation is that any emerging issues will require minimal input from the UX designers.

Advocates for this approach include Cooper [1999], Constantine [2002] and Beyer et al. [2004]. The advantages they cite include a holistic approach to UX design: "...the overall organization, the navigation, and the look-and-feel — must be designed to fit the full panoply of tasks to be covered" [Constantine 2002]; concentrating expertise within each discipline: "The great strength of agile methods is that they focus the engineers on doing what engineers do best" [Beyer et al. 2004]; and that a separate UX design phase saves costs by determining the correct solution before costly changes in the implementation phase are required: "Iterating in construction just throws your money away."[1]

---

[1] From Alan Cooper's keynote at the Agile 2008 conference, which can be accessed at http://www.cooper.com/journal/agile2008/.

8.2.2 Study 2: Valuing togetherness

Team2A and Team2B's relaxed, on-going interactions contrasted with the systematically planned activities that prompted the conversations between the developers and UX designers of Team1. During the course of the day, UX design issues were generated and dealt with alongside technical implementation issues. Breaking up and prioritising the UX design work was no more remarkable to Team2A and Team2B than carding and prioritising the non-UX related work. The UX designer was continually aware of the implementation issues of their designs and the developers were constantly aware of UX design decisions. Every role on the project was involved in UX design discussions, which led to decisions that took into consideration both design values and technical constraints.

In the case of Team2A and Team2B, the managing director of the organisation informed us that he hired individuals who could fulfill the responsibilities of their main role, but who were also able to contribute outside their roles, or who could learn to do that. In light of the managing director's decision and the practice of Team2A and Team2B, we gain insight into another view on how UX design and Agile development should be combined. This can be expressed in terms of valuing the Agile and UX disciplines working as closely as possible.

Instead of pitting UX design against construction, there is an emphasis on the contributions of various roles and their continuous interaction throughout the project. Valuing working closely together is based on the assumption that each role has a valuable contribution to make to the overall product. If this were not the case, Team2A and Team2B may have engaged in activities where boundaries between roles and between disciplines were maintained. This view acknowledges that UX design issues, along with technical issues, emerge throughout the project and are dealt with via the ongoing interactions between all the roles on the project.

Those who support this view promote favourable environments in which project stakeholders continuously interact in ways that are helpful in progressing with their

work — through consideration of the team as a whole irrespective of the skills of the individuals, or through enhancing collaboration. Beck and Andres [2004] include the practice *Whole Team* in eXtreme Programming: "Include on the team people with all the skills and perspectives necessary for the project to succeed." Highsmith [2009] connects collaboration between roles with project success: "Collaboration, not eliminating specialty roles — like business or product specialists — is the key to success." Miller [2005] has provided a process model in which UX design work is interleaved with implementation work throughout the duration of the project, thereby ensuring input from the UX design role throughout.

8.2.3   Study 3: Valuing learning

The different circumstances of Team3, i.e., the trial status and early stage of development of their project, must be taken into account when considering the activities that were observed. The team displayed a positive attitude, recognising the "experimental" and improvisational character of the work, by comparing their trial experiences with how they would "normally" work. With Team3, the work of the designers and developer were combined via their discussions. As each individual went about their work, they asked for input from the other team members as and when required. Input from the developer or designers became part of the working code or designs without the need for intermediate activities.

Section 6.5.2 previously mentioned how the organisational setting in which Team3 were embedded differed from the organisation of Team1, in that they displayed a willingness to try something different. As an organisation, they had taken on a project which was going to be different to any other project they had previously been involved in, they were attempting a new project management strategy, and they were bringing the developer and designers into one room. Such decisions could not have been trivial to carry through, and required deliberate decision-making. These decisions were in the interests of learning something new about ways of working together.

Before this point, the Head of UX had spent a year bringing awareness of Agile methods to the organisation and building up trust and enthusiasm for a change. Eventually circumstances were conducive to conducting the trial. According to the discussions in the "Agile room", Team3 had similar experiences in their daily work to those of Team1 before the starting the trial. However, the organisation agreeing to the trial and agreeing to undergo some upheaval in conducting the trial suggests that there was probably less confidence among members of the organisation that this was the best way to create software. The Head of UX suspected that working closer together would bring improvements and went about implementing that idea in the form of the trial we observed.

Without the wider organisation endorsing the approach that the Agile and UX disciplines could learn to work closer together, they would have been less likely to agree to the trial. It can also be assumed that if separation between the UX designers and Agile developers was as important to this organisation as it was to the organisation in Study 1, then this would also have made the trial less likely. In valuing learning to work closer together an underlying assumption may be that Agile development and UX design exist independently, however, a closer relationship can be achieved by way of learning (by doing). As mentioned in section 6.5.2, another underlying assumption here is that design issues that arise during implementation can potentially be problematic and need to be dealt with during the early stages of the project, and that the developer's input into early design decisions mitigates the risk of serious implementation issues arising later.

There is evidence in practitioner accounts that the combination of Agile development and UX design in practice is refined over the course of time. A certain approach to the combination is attempted, and then changes are made based on those experiences. Lindström and Malmsten [2008] state that the combination of iterative development with UCD "evolved" during the project. The account by Broschinsky and Baker [2008] tell of ongoing improvements in how the Human Factors team worked with XP developers. Singh [2008] found that even with usability experts on Scrum

teams, maintaining usability as a priority "on par with the underlying functionality" is problematic. Therefore, they added new role focusing on usability issues — the usability product owner.

What the reports by Lindström and Malmsten [2008], Broschinsky and Baker [2008] and Singh [2008] have in common is not only a commitment to continually improving how they work together, but also an acknowledgment of the key role of management in enabling developers and designers to learn and improve how they work together. Lindström and Malmsten [2008] point out that management had to be "convinced" to work in this way: "Since there is no final specification of requirements to show up front, before the project starts, you have to convince management that the methodology and process themselves will lead to a good product." Singh [2008] reports that senior management identified an appropriate project to experiment with adding a usability product owner, while in the report by Broschinsky and Baker [2008], human interaction became a very high priority for the organisation and "Human Factors was supported at the very top of the company." Similarly to the teams described in sections 8.2.1 and 8.2.2, environments in which learning to work together can take place emerge as a result of significant management decisions.

8.2.4 Implications of surfacing values and assumptions

In some respects the teams in all three 'studies seem rather similar: they were embedded in organisations that recognised the combination of UX design and Scrum as essential for software development, their organisations highly valued UX design and were successful at delivering software. Yet, despite these similarities their experiences of practice were different. Practice was shaped by decision-makers external to both teams, around values and assumptions concerning how best to create quality software. Team1 was developing software amid values promoting separation between UX design and Agile development disciplines, manifesting as a systematic, separatist approach to integrating and coordinating the two disciplines. Team2A, Team2B and Team3,

were developing software amid values bringing UX design and Agile development closer together, manifesting as a subtle ongoing effort promoting shared awareness of UX design values and technical constraints, as well as shared decision-making. In the case of Team1, the developers were taking direction from the UX designers whereas in the case of Team2A, Team2B and Team3, the various roles were negotiating what that direction should be.

The resonances of these views in the existing literature demonstrates that the values and assumptions shaping the practice of the teams are not only peculiar to these teams. Rather, the values and assumptions underlying decisions to separate or keep the Agile developers and UX designers together, (in)form the wider ongoing debate about how Agile development and UX design should work together. The debates concerning Agile development and UX design have been outlined in section 2.4 and there are clearly echoes of those debates in what the analyses of Study 1, Study 2 and Study 3 have shown to shape practice in these teams. In particular, there are echoes with the debate around the place of design and construction.

Section 8.4 presents the integration of Agile development and UX design as a multidimensional concept. Those teams working amid similar values and assumptions about how best to create quality software, also have similar experiences of integrating UX design and Agile development. That is, the separation characteristic of Study 1, sees their experiences integrating UX design with Agile development as discordant with the experiences of integrating Agile development with UX design that characterised Study 2 and Study 3. Although the details of the work of the teams in Study 2 and Study 3 varied, subscribing to similar values concerning how best to develop quality software, resulted in their experiences aligning.

8.3 Comparing the field work with the accounts of practice

The analysis of the accounts of practice presented in chapter 7 found that while some claims of how Agile development and UX design is combined converge, there

remains a messiness around our understanding due to the unresolved conflicts in the accounts. Considering the findings of the field work in chapters 4–6 and the understanding of Agile development and UX design integration in practice related in the preceding discussion, this section compares the findings from the analysis of the published accounts of practice with the findings from the field work. What has emerged from the analysis of accounts of practice differs in important ways from the analysis of the field work.

8.3.1 Combination strategies

Section 7.5 presented the combination strategies that emerged from accounts of practice — merging, inserting and adapting. These strategies are employed in accounts by practitioners as illustration of how they combined Agile development and UX design. These strategies have as their aim maintaining focus and coordination, which relies on casting the work carried out by the Agile developers and UX designers as discrete tasks that in the right order, and in the right combination, achieves the integration of Agile development and UX design. Section 8.4.1 explained that the analysis of the field work moved away from this view on integration, to attend to the work that occurs on a moment-by-moment basis and the social processes that made the integration of Agile development and UX design possible.

8.3.2 Agreement, conflict and contingency

Section 7.6 presented how the accounts regarding the combination of Agile development and UX design in accounts of practice contain agreements, conflicts and examples of contingency. In accounts of practice, there is agreement among practitioners that combining Agile development with UX design is a positive thing and something better than what they had been doing before (see section 7.6.1). The participants in Study 3 were vocal in their endorsement of working together in one room and the benefits of being able to show the client working software during the

development process (see section 6.4.1). This was new and remarkable to them. The participants in Study 1 and Study 2 did not express similar sentiments — in the case of Study 1, the developers were working with UX designers who were not committed to the deadlines of the developers. In the case of Study 2, the participants did not find the combination of UX design and Agile development in any way remarkable.

Sections 7.6.2 and 7.6.3 presented the conflicts and tensions that emerged from accounts of practice and the disagreements between accounts. Statements such as "Agile development and UX design are a natural fit" (a similar statement appearing in the account by Williams and Ferguson [2007]) was found to have little relevance in practice, as evidence suggests that whatever the differences in the methods may be in the texts that describe them, practitioners work through any difficulties as they arise on a day-to-day basis. Agile developers and UX designers do what is required in order to get their job done, by maintaining focus and coordination between their tasks, expecting certain behaviours from others, maintaining mutual awareness, negotiating progress and engaging with each other. The participants in Study 1 did experience tensions between the Agile developers and UX designers, in particular, frustrations similar to that reported by Frank and Hartel [2009] in section 7.6.3 — that the Agile developers were frustrated with not being included in design decisions. However, the participants in Study 2 and Study 3 resolved tensions by engaging frequently and managing each other's expectations. The empirical evidence presented in this dissertation suggests that the conflicts and tensions that do exist in practice do not arise from the Agile development method, or the UX design method, but from the contingent use of these methods *in a setting*. That is, whether or not the integration of Agile development with UX design is fraught with frustration depends on the circumstances in which they are being integrated, and the extent to which the wider organisation in which the teams are embedded support or hinder their integration.

8.3.3 Similarities and differences

Section 7.8 showed how empirical studies in the literature take a particular interest in the similarities and differences between Agile development and UX design. One similarity cited in accounts of practice is that of the iterative nature of UX design and Agile development and the example given was from the account by Hussain et al. [2009b]. In light of the empirical evidence presented in this dissertation, the iterative nature of UX design and Agile development was unable to overcome the separation between the Agile developers and UX designers in Study 1. The iterative nature of the methods did little to address the UX designers' lack of engagement with the Agile developers. In Study 2, the creation of the design and the code emerged in a synchronised manner. While the iterations were likely useful for allowing both the design and the code to be iterated on, their iterative nature was not the key to understanding Agile development and UX design integration in that setting. In Study 3, the UX designers and Agile developer were working on different aspects of the software under development and so the iterative development of the code and design were not synchronised. Therefore, whether the similarity which is claimed in the empirical research literature to aid the integration of Agile development and UX design actually does so in practice, depends on the circumstances in which they are being integrated.

A similar argument can be constructed for the differences upheld in the empirical research literature. That Agile development and UX design have different provenances and traditions that challenge practitioners has been recognised in section 2.3. However, the empirical evidence presented in this dissertation suggests that in practice, workflow is maintained and the developers and designers continue to make progress in the ways that are discussed in the next section.

8.4 Achieving integration

The findings from the analysis of accounts of practice in chapter 7 combined with the findings from the field studies (chapters 4 6) allows for a closer look at how the *integration* of Agile development and UX design has been addressed in accounts of practice and achieved in practice. The findings have highlighted how integration is complex, situated and, far from a spontaneous occurrence, depends on considered, deliberate decisions to engage other individuals. This section brings the findings from accounts of practice and field studies together by drawing on five themes that were identified as essential to achieving the integration of Agile development and UX design: focus and coordination, expectations about acceptable behaviour, mutual awareness, negotiating progress, and engaging with each other.

8.4.1 Integration as focus and coordination

As identified in analysis of the accounts of in chapter 7, practitioner accounts give primacy to maintaining focus and coordinating between tasks when integrating Agile development with UX design. Focus here refers to the dual aims reported in accounts of practice, that of a focus on usability and on releasing working software. This view on integration relies casting the development of software as discrete tasks, practices and tools, that in the right combination achieves the desired focus. Coordinating between the tasks, practices and tools is a related concern. The nature of the tasks lend themselves to some ordering. That is, the work of understanding the user is typically carried out in advance of developing the code, so that the results can be incorporated into the coding. The issues around the timing and scheduling of these tasks has been found to be non-trivial and was addressed in section 2.3.1.

In Study 1, a focus on integration as the result of discrete tasks, referred to collectively as "integration activities", was presented in chapter 4. However, in Study 2 and Study 3, the data analysis did not result in comparable activities. The way in which the teams in Study 2 and Study 3 went about their work was not amenable

to this view on integration. The discrete integration activities carried out by the participants in Study 1, were the inventive way the developers elicited the kind of interaction from the UX designers they needed in order to progress in their work. The integration activities can be seen as a response to the culture of separation between the Agile developers and UX designers prevalent in the organisation in which the developers and designers were embedded. This separation was not maintained in the organisations in Study 2 and Study 3 and therefore the integration between Agile development and UX design was achieved in other ways.

8.4.2 Integration as expectations about acceptable behaviour

The work of integrating Agile development and UX design is shaped by expectations about how the other group should support them in their work. These expectations in turn are shaped by a complex interplay of organisational- and team-level commitments that divide responsibilities between developers and designers, and establish the rhythms of work for each group. The analysis of Study 1 identified the impact of differing expectations about acceptable behaviour by the Agile developers and UX designers. The developers were responsible for delivering working software, which adhered to the UX designs handed to them, and to develop that software within a set deadline. Due to segregated responsibilities and the iterative, incremental rhythms of Scrum, the developers in Study 1 relied on the timely delivery of the UX designs in order to progress with transforming the paper-based UX designs into working software. Thus, the developers expected that the UX designers would be willing and able to accept feedback, produce a redesign every Sprint based on that feedback, and would continue doing so until the release date. The UX designers were engaged in work (producing paper-based wireframes and visual designs) that could progress without requiring input from the developers. The organisation discouraged the UX designers from concerning themselves with the implementation issues of their designs. Further, the UX designers were involved with many different teams across different projects and expected to hand over their designs to the Agile developers, and move on to the

next project. The UX designers were therefore not dependent on the developers in the same way and were not expecting to have further discussions with the developers about the designs that had been handed over. These mismatching expectations about acceptable behaviour in Study 1, led to wasted planning efforts and developers working over-time to meet deadlines.

In Study 2, the frequent interactions between the developers and designer ensured that expectations about how the developers and designer should support each other in their work was negotiated on a day-to-day, moment-by-moment basis. Changes in either the UX designs or the code could be communicated to the team immediately and the delays and disruptions to work rhythms evident in Study 1, were not experienced by the Study 2 participants. The UX designer was not surprised with developer feedback in the way the UX designers were surprised in Study 1 (§ 4.4.4). That issues would be discussed by everyone involved was part of the normal course of the day.

The expectations about acceptable behaviour as found in Study 3 has been discussed in section 6.5.2. The normal expectations about Agile developer and UX designer interactions in Study 3, were adjusted in the trial situation. Working together in one room made them available to each other in ways that had not been experienced outside the trial situation, such that their dependencies on each other could be negotiated on a day-to-day, moment-by-moment basis. The negotiations between the developer and designers at this early stage of their project were shaping dependencies between them that would come to bear in later stages of the project when the UX designs are turned into working software. Similar to Study 2, the developer was not surprised about being asked questions by the UX designers and vice versa. The developer and designers made themselves available to be part of the discussions throughout the observed period.

8.4.3 Integration as mutual awareness

The work of integrating Agile development and UX design depends on a mutual awareness between the developers and designers, specifically the UX designers being aware of what constitutes work for the Agile developers and the Agile developers being aware of what constitutes work for the UX designers. The level of awareness the developers and designers have of each other's work affects their ability to make informed judgments around coordinating their work.

The developers and designers in Study 1 were displaying a lack of awareness of each other's work arrangements. The UX designers were missing deadlines, unaware of the deadlines imposed by the sprints, and were unavailable when developers required their input into design decisions. The developers were unaware that the UX designers were moving offices and could not find a designer for making decisions. Some decisions around design changes that showed up in the designs delivered to the developers had not been communicated to the developers, leaving the developers unaware of the rationale behind certain design changes and, consequently, with more questions. The Agile developers' need to gain approval from the UX designers was driven by the rigid role boundaries that were maintained   designers made design decisions and developers made coding decisions.

By contrast, the designer in Study 2 was continually aware of the implementation issues of the designs and the developers were constantly aware of UX design decisions. The QA role was instrumental in highlighting the issues that the developers and designer needed to discuss. As mentioned in section 8.4.2, the developers and designers were available to each other for questions as they arose and were frequently engaged in discussions, thereby creating awareness and sharing decision-making responsibilities. Similarly to the designer and developers in Study 2, the designers and developer in Study 3 were creating awareness and sharing decision-making responsibilities by frequently engaging each other in discussions. The fluid role boundaries characteristic of the teams in Study 2 and Study 3 served to enhance mutual awareness, whereas

the rigid role boundaries between the Agile developers and UX designers in Study 1 served to hamper mutual awareness. In Study 1, schedules and progress were impeded by this lack of mutual awareness, whereas, the developers and designers in Study 2 and Study 3 did not experience such impediments.

8.4.4   Integration as negotiating progress

The day-to-day work of integrating Agile development and UX design contends with various uncertainties around client expectations and requirements. Those with the appropriate knowledge who can dispel these uncertainties (such as clients, or even UX designers) are not always present when decisions need to be made. Despite this, the teams in all three studies continued to move forward, avoiding a complete breakdown in the workflow. Progress was negotiated among the developers and designers, however, the values endorsed in the settings of the teams served to either constrain or enable the developers and designers in negotiating progress.

The constrained manner in which the Agile developers and UX designers in Study 1 integrated their work can be attributed to the reluctance with which the developers made design decisions. The separation that was maintained between the developers and designers in this study hindered their progress by obscuring the realities of what constitutes work for one group, from the other. The organisation was committed to developing quality software, by placing a higher value on the UX designers' input.

Section 5.4.3 noted that in Study 2, during the course of the day, UX design issues were generated and dealt with alongside technical implementation issues. Breaking up and prioritising the UX design work was no more remarkable to the teams than carding and prioritising the non-UX related work. Every role on the project was involved in discussions that promoted consideration of design values alongside technical constraints. The effect was the largely indiscernible (to the observer) nature of UX design issues — they simply formed part of the work that was discussed and carried

out. There was no power hierarchy for the participants in Study 2 to contend with, as there was in Study 1, which allowed issues to be resolved as part of the ongoing flow of work, irrespective of who had contributed to resolving the issue.

The team in Study 3 were making progress despite lacking specificity from their client about what had to be done next. Section 6.5.1 discussed how the developer and designers were negotiating progress amid this uncertainty. Progress was negotiated among the developer and designers, who discussed what needed to be done by referring to meetings with the client, the existing website and some information on the client organisation (e.g. management structures). The developer and designers were negotiating the way forward by asking and answering questions, keeping each other informed of decisions, and involving each other in decision-making. As with the participants in Study 2, everyone's input was considered valuable and no one side was dictating the direction to be taken with the software.

8.4.5 Integration as engaging with each other

The empirical evidence presented in this dissertation showed how developing software was interactively accomplished between the Agile developers and UX designers. Integrating Agile development with UX design in practice, relied on recurring efforts in engaging other individuals for their input. The Agile developers and UX designers worked on their own tasks but required the involvement of those with decision-making power, or expertise, when the work demanded it. There are therefore two modes of work of interest in Agile development and UX design integration. One is common action, described in section 6.4.2: the work the developers and designers do collectively when they are focusing on the same part of the software together (synchronously). An example of common action is the discussion that ensues when either the developer or designer raises an issue and calls everyone's attention to a specific design element or client requirement. The other mode of interest is when the Agile developers and UX designers are focusing on their own tasks, doing individual work

(asynchronously). For the Agile developers and UX designers in all three studies, accomplishing their work relied on switching back-and-forth between these modes with articulation work [Strauss 1988], both explicit and implicit, coordinating their switching.

The Agile developers engaging with the UX designers in Study 1 required explicit articulation work, including discussions around why the UX designers' engagement was necessary. Had the developers in Study 1 not approached the UX designers, there would have been no way for the UX designers to know that their input was required. The coordination of the Agile developers' work and the UX designers' work had to be explicit to overcome the lack of mutual awareness between developers and designers.

Articulation work in Study 2 was of a more implicit nature, where the Agile developers and UX designer understood what needed to be done next without requiring extensive discussions between them and certainly no discussion around why the designer should be involved. The higher level of mutual awareness between the developers and designer than was present in Study 1, ensured ongoing engagement understood by all as necessary for working through issues as they arise.

In Study 3, articulation work was instrumental in the (re)creating of progress as the team contended with day-to-day uncertainties. In this study, articulation work was not required for encouraging the UX designers to be involved (as in Study 1), or to explain why they should be involved, as much as it was about status updates and clarifications in the interest of workflow. As with Study 2, the frequently recurring moments of engaging with each other helped the developer and designers coordinate their work and decide what further action was meaningful with respect to the software.

8.5   Limitations

Generalising from the findings of this research should be approached with care. It has to be acknowledged that the teams studied in this dissertation are not considered to cover all the possibilities found in practice. There are many more approaches to Agile

development than Scrum and the inventive and creative ways in which designers work present considerable variety in what can be found in practice. However, the teams studied are considered to not be atypical of teams in practice.

The findings in this dissertation emphasise the situated nature of combining Agile development with UX design and in the extreme view this may imply that predicting actions in other settings is impossible. The aim of this research is not to predict practice in every possible setting. Rather, the aim with this research has been to find how developers and designers achieve their work, and in that way uncover mechanisms that may persist across various settings subject to further investigations into practice.

This research has shown the importance of contextual values and their role in shaping practice. However, this is only one aspect of setting that may have a shaping role. This research demonstrates the kinds of findings that can link settings with the Agile development and UX design work that unfolds within them, providing an alternative perspective on what is rarely acknowledged to be a complex, multi-faceted challenge.

8.6 Summary

In practice, experiences of Agile development and UX design integration align with values (concerning how best to develop quality software) endorsed by the organisations. Two views emerged from the analysis of the field work and there is support in the literature for both: (1) the best way of developing quality software is by keeping the Agile developers and UX designers separate, or (2) the best way of developing quality software is via the Agile developers and UX designers working closely together. The organisation in Study 1 was the only organisation of the three studied to subscribe to separating the Agile developers and UX designers and maintain rigid boundaries between their roles. Thus, integrating Agile development with UX design was associated with working to overcome the separation between the developers and designers.

The claims identified in the accounts of practice were compared with the findings from the results of the field work. Whether the claims in the accounts of practice regarding the integration of Agile development and UX design hold up in practical situations, depends in important ways on the circumstances in which they are being integrated. The empirical evidence regarding integration suggests that Agile developers and UX designers do what is required in order to get their job done. In practice, workflow and progress is maintained by maintaining focus and coordination between their tasks, expecting certain behaviours from others, maintaining mutual awareness, negotiating progress and engaging with each other.

CHAPTER 9

Conclusion

9.1   Final summary

The findings in this dissertation allow us to answer our overarching research question: *How are UX design and Agile development combined?* It has been argued in this dissertation that integrating Agile development and UX design is an on-going achievement in practice. The major themes that describe how integration was achieved, emerged from an investigation into published accounts of practice found in the literature, as well as an investigation into teams in organisational settings. A better understanding of how Agile developers and UX designers work together is motivated by the confused state of accounts in the literature and the limited discussion on how the combination of Agile development and UX design is shaped by practical settings. The perspective taken in this dissertation, considers combining Agile development and UX design as a challenge that plays out in practice. Understanding how they are or can be combined requires an understanding of how the integration is achieved by people, engaging in Agile development and UX design practice.

The sub-questions posed for the field work was *What shapes the combination of Agile development and UX design in practice?* and *How is Agile development and UX*

design work accomplished? In the studies of day-to-day practice, explaining how the Agile developers and UX designers working together did not orient to descriptions of processes decided outside the circumstances in which they were combined, neither was practice a disorganised muddling. Integrating Agile development and UX design depended on the day-to-day negotiations between the developers and designers in the specific circumstances of each setting. Achieving integration depended on contextual values held in the organisation, with regards to the combination of Agile development and UX design. That is, the work of integrating UX design and Agile development in each practical setting was shaped by values embedded in the organisations. The different characteristics that work assumed in the different settings, required a shift in the observation and analysis approach of the researcher — from a task-oriented focus to a focus on talk and interactions between individuals.

The sub-question posed for the analysis of the published accounts of practice was *What are the existing perspectives on combining Agile development and UX design in accounts of practice in the literature?* The analysis of the published accounts of practice has illustrated how empirical studies and practitioner reports include different assumptions about practical settings and do not emphasise the differences between Agile development and UX design in the same way. Empirical studies would seem to suggest that the activities of the UX designers and Agile developers can be slotted together in an unproblematic way, giving little attention to the status of Agile development and UX design in practical settings. Whereas practitioner reports contain many contingencies on which a successful combination of Agile development and UX design depends. It is evident from the analysis of the published accounts of practice that there is a disjointedness in how empirical studies and practitioner reports portray the differences between Agile development and UX design, and the status of UX design and Agile development in organisations. The analysis of the published accounts of practice found an emphasis on processes, methods and tools.

Combining the findings from the analysis of accounts of practice in the literature and the field work, five themes were identified as essential to achieving the integra-

tion of Agile development and UX design: (1) focus and coordination, (2) mutual awareness, (3) expectations about acceptable behaviour, (4) negotiating progress and (5) engaging with each other. These are what are referred to in this dissertation as "the facets of integration." The extent to which these facets are supported or hindered, and therefore the extent to which integration is enabled, depends on contextual values concerning the combination of Agile development and UX design embedded in the wider context (i.e., the organisation). As the problems posed by crossing disciplinary boundaries are worked through, there is a need for a better awareness of the work settings in which the integration of Agile development and UX design unfolds. Instead of waiting for one ideal approach to emerge and successfully translate to any other work setting, attention should be focused on the work cultures involved in UX design and Agile development in practice, and understanding and supporting the mechanisms of the work cultures that *enable* the integration of Agile development and UX design.

9.2 Contributions

The contributions of this research come from an independent analysis of published accounts of practice and the analysis of the field work, whose findings have been compared and synthesised in order to answer the research questions. The analysis of published accounts of practice and the field work also provide contributions in their own right, as set out in this section. The synthesis provides the main contribution of this research:

> Agile development and UX design **integration is an on-going achievement in practice**. The findings in this dissertation show how integration is achieved by practitioners through maintaining:
>
> 1. **focus and coordination** — The analysis of the published accounts of practice found that the work of the Agile developers and UX de-

signers are combined in order to focus on usability and on releasing working software. Integrating the work is achieved by coordinating between the tasks of the developers and designers and establishing the right sequence. The accounts of practice present Agile development and UX design integration as merging processes, inserting and adapting methods, tools and techniques in the software development effort.

2. **mutual awareness** — The work of integrating Agile development and UX design depended on a mutual awareness between the developers and designers, specifically the UX designers being aware of what constitutes work for the Agile developers and the Agile developers being aware of what constitutes work for the UX designers. The findings from the field work identified that a lack of awareness of each other's work arrangements contributed to problems such as missing deadlines and delays in resolving questions. Whereas, in the studies where mutual awareness between the developers and designers was maintained, questions were resolved as they arose, discussions were frequent, and decision-making responsibilities were shared.

3. **expectations about acceptable behaviour** — The work of integrating Agile development and UX design was shaped by expectations about how the other group should support them in their work. The findings from the field work identified how a mismatch in expectations about acceptable behaviour contributed to wasted planning efforts and working over-time to meet deadlines. Matching expectations, as a result of negotiations on a day-to-day, moment-by-moment basis, prevented delays and disruptions to work rhythms.

4. **negotiating progress** — The day-to-day work of integrating Agile development and UX design contends with various uncertainties around client expectations and requirements. The findings from the

field work showed that progress was maintained through the interactions between the developers and designers. Progress flowed when every role on the project was involved in discussions, could contribute to decision-making and resolve issues as part of the on-going flow of work. Progress was constrained when decision-making responsibilities were segregated, and the separation between developers and designers obscured the realities of what constitutes work for one group, from the other.

5. **engaging with each other** — Integrating Agile development with UX design in practice, relied on recurring efforts to engage other individuals with decision-making power, or expertise, as and when the work demanded it. The discussion in this dissertation has relied on the articulation work between the developers and designers. The field work found that the articulation work had various aims in each setting, depending on what the developers and designers required from each other in order to get their work done. Where the developers and designers were separated, the explicit articulation work was aimed at eliciting the support from the designers that was required by the developers. Where the developers and designers were working closely together articulation work sustained the on-going engagement between the developers and designers such that the workflow was maintained.

The field work provides the following contributions:

The work of the developers and designers depends on the values endorsed by the organisations in which the developers and designers are embedded. The facets of integration (1–5 above) are **shaped by values embedded in the settings in which practitioners work**. Analysis of the field work highlighted the role of contextual values concerning how best to de-

velop quality software by combining the skills of the Agile developers and UX designers. Two views emerged: (1) the best way of developing quality software is by keeping the Agile developers and UX designers separate, and (2) the best way of developing quality software is via the Agile developers and UX designers working closely together. Those teams embedded in settings endorsing similar values, also had similar experiences in carrying out their work. Agile development and UX design integration explained as an on-going achievement in practice, along with the role of contextual values in shaping practice, addresses the current lack of understanding of the day-to-day work that integrates Agile development and UX design via:

1. **Accounts of the combination of UX design and Agile development in work settings.** This research generated accounts based on observation and engaging with participants in their work settings, concentrating on how the day-to-day work of Agile developers and UX designers are achieved. While spending time with participants in their work settings, the researcher could access the naturally occurring conversations, the rhythms of the work day, scheduled meetings and other events that developers and designers face in their everyday work. This made it possible to build an account of how Agile development and UX design is integrated that tends to the variation in practice across settings and is not limited to descriptions of processes, methods and tools.

2. **A perspective on practice as situated action.** This research tends to the localised and contingent nature of the work, and work as shaped by the practical setting. The interactions between individuals and the meanings they assign to their interactions, became the central concern. The improvised, emergent course of action was negotiated by the developers and designers in their purposeful inter-

actions with each other. Coordination and integration of the work of the UX designers and Agile developers were achieved through these on-going negotiations of order. This presents an alternative to current approaches to combining Agile development and UX design that focus on processes, methods and techniques.

The analysis of the published accounts of practice provides the following contributions:

Within each group of accounts — the practitioner accounts and empirical study accounts — certain claims about combining Agile development with UX design converge (e.g. the combination strategies in the practitioner accounts and the treatment of the differences between Agile development and UX design in the empirical study accounts). There also is a coherence within each group concerning the type of challenge each considers the combination to be. In the practitioner accounts and empirical study accounts the concerns are similar: getting UX designers and Agile developers to work together requires effort and support in order to achieve their aims. The practitioner and empirical study accounts agree that addressing the problems that combining Agile development with UX design brings, relies on finding the right process, tools and methods. Practitioner accounts present processes, tools and methods for maintaining focus and coordinating between tasks when integrating Agile development with UX design, while accounts in the empirical studies present processes, tools and methods for overcoming the differences between Agile development and UX design. However, **unresolved conflicts** remain within and between these groups. This analysis has highlighted where the discourse regarding practice becomes disjointed: in the treatment of the differences between Agile development and UX design, and the assumptions about the status of UX design and Agile development in organisations. The need for engaging with practitioners in the settings in which they work emerged

from:

1. **A comparison of accounts from practitioner reports and empirical studies.** The analysis of accounts of practice distinguished between accounts from practitioner reports and accounts from empirical studies. The distinctions maintained during the analysis allowed the different assumptions in the accounts to emerge. This distinction has not been made by the current Agile/UX literature. Instead, accounts from practitioners and accounts from empirical studies are treated as the same, which obscures the different assumptions on which the accounts are based. This is problematic for developing an understanding of how Agile development and UX design are combined in practice.

2. **Unanswered questions regarding setting.** After an in-depth analysis of the published accounts of practice, it was found that the accounts have little to say about how work settings shape the combination of Agile development and UX design in practice. The themes emerging from the analysis in this chapter do not explain what shapes the combination of Agile development and UX design in practice. Neither do they explain how Agile development and UX design work is accomplished on a day-to-day basis.

9.3 Future work

Future work could extend the research in this dissertation in the following ways:

1. The scope of the field work carried out in this dissertation can be extended by studies involving different settings and different team arrangements. The field work for this research was conducted with Agile developers using Scrum and with Agile developers and UX designers inhabiting the same organisations.

The claims from this research could be investigated further in organisations using different Agile development approaches (e.g. Feature-Driven Development). Further investigations could include Agile developers and UX designers who work on common projects, but are part of different organisations. The generalisability of the claims in this dissertation could be improved if the facets of integration and the values shaping them extend to other settings and other team arrangements. Alternatively, different sets of values may arise from the different settings and the different team arrangements that can contribute to a fuller picture of how Agile development and UX design integration is achieved in practice.

2. The developers and designers studied as part of this research were developing software for either the web or mobile devices. The developers and designers developing for the mobile in Study 2, were working very closely together. When this was fed back to this team in the feedback session, the manager reflected on why working closely together might be related to developing for the mobile, rather than for the web:

> "A lot of the detail of a mobile UI [user interface] would actually be stuff that happens over time. So I can't give you a fixed screen and say "Go build this." It's as much about behaviours and how things react [...] Because that would then bring design and development much closer together — if it's about behaviours as much as it is about visuals."

Further investigations into how Agile development is combined with UX design could investigate whether the type of product under development influences the interactions between developers and designers.

3. This research has found that contextual values and assumptions have a role in shaping Agile development and UX design practice. One way of taking this discussion further, could be to analyse practice in terms of *culture*. The discussion

in this dissertation has referred to values, behaviours and assumptions, which can be considered as elements of a culture. Cultures endorse and constrain the behaviours of individuals embedded in organisations [Vaughan 2002] and may therefore provide a useful analytical lens for studying Agile development and UX design in practice. Vaughan [1996] has shown that culture does not necessarily have to extend to an entire organisation. By studying *work group culture* the focus can be the "set of solutions produced by a group of people to meet specific problems posed by the situation that they face in common" [Vaughan 1996, p. 64]. This can be taken up in studies of the literature to examine how the values and cultures endorsed by the Agile development communities and UX design communities inherit from those endorsed by the wider HCI and SE communities.

4. The analysis of the published accounts of practice in this dissertation found that processes, tools and methods were of interest to the Agile/UX community. The analysis pointed out the lack of understanding of how Agile development and UX design is combined in practice that brings the utility of creating more processes, tools and methods into question. However, based on the findings in this dissertation, how processes, tools and methods participate in social processes could lead to useful insights into Agile development and UX design practice, as well as inform further development of processes, tools and methods. That is, further investigations into how Agile development and UX design are combined can investigate what processes, tools and methods are useful for sustaining the facets of integration and how processes, tools and methods feature in the interactions between Agile developers and UX designers in practice.

APPENDIX A

Appendix to Chapter 7

A.1 Grey literature

Type	References
Tutorial descriptions/reports	[Hudson 2003; Patton 2004; Constantine and Lockwood 2003]
Panel descriptions	[Sharp et al. 2006; Lukanuski et al. 2008; Federoff et al. 2008]
Workshops	[Martin et al. 2008; Sy and Miller 2008]
Demonstration	[Hosseini-Khayat et al. 2009]
Keynote	[Maurer 2009]
SIG	[Miller and Sy 2009]
PhD consortium	[Lee 2006]
Poster	[Keenan et al. 2009]
Position paper	[Kane 2003]
Total grey literature	14

A.2 Excluded literature

Excluded at Stage 1:

Reason	Frequency
grey literature	14
non-English	2

full texts unavailable	3
not peer-reviewed (academic dissertations, columns, white papers)	8
publications generated by the research in this dissertation	3
subtotal	30

Excluded at Stage 2:

Reason	Frequency
no details of practice in context	9
not enough details of practice in context	3
review of the literature	2
different focus	37
book substituted with a shorter publication	1
Total excluded	82

APPENDIX B

Ethics documents

HUMAN PARTICIPANTS AND MATERIALS
ETHICS COMMITTEE (HPMEC) PROFORMA

Please complete and send to:

John Oates (j.m.oates@open.ac.uk), Chair,
Human Participants and Materials Ethics Committee (HPMEC)
Centre for Childhood Development and Learning (CHDL),
Briggs, Walton Hall, Milton Keynes
Also send a copy to Research-ethics@open.ac.uk

If you have any queries before you fill in this form please look at the
Research Ethics (intranet) web site: **http://intranet.open.ac.uk/research/ethics/**

Title of project
A short, descriptive title.

Interaction design and Agile software development practice.

Schedule
Time frame for the research and its data collection phase(s).

A two week observational study in September 2008. The exact dates will be determined
in consultation with the [organisation], the company that provides access.

Abstract
A summary of the main points of the research, understandable by a non-specialist.

The work is concerned with exposing a research student to the reality of Agile software
development and the challenges of observing and recording professional practice. This
is part of the student's postgraduate training in research methods, prior to an extended
period of (similar) observational work as part of her main PhD research. The
development team at the [organisation] are of interest due to their integration of
interaction design techniques and agile development practices. Apart from providing
skills training, the team's work practices fit the interests of the research student, making
them suitable for further studies.

Source(s) of funding
Details of the external or internal funding body (e.g. ESRC, MRC).

Research studentship.

Justification for research
What contribution to knowledge, policy, practice, and people's lives the research will make?

The student will gain a valuable understanding of the reality of interaction design and Agile software development, and the challenges of observing and recording professional practice. This understanding will later be put to use with more sustained observational work, ultimately making a contribution to knowledge in the area of interaction design and Agile software development. This will lead to better, more appropriate software for users.

Investigators
Give names and units of all persons involved in the collection and handling of individual data. Please name one person as Principal Investigator (PI).

Helen Sharp (Principal supervisor), Hugh Robinson (Co-supervisor)), Jennifer Ferreira (Research student).

Published ethical guidelines to be followed
For example: BERA, BPS, BSA (see Research Ethics web site for more information).

BPS.

Location(s) of data collection
Give details of where and when data will be collected. If on private, corporate or institutional premises, indicate what approvals are gained/required.

The [organisation] have offered access to an agile development team based at [address] so that the research student may observe, and gain a better understanding of, professional software development practice.

Participants
Give details of the population from which you will be sampling and how this sampling will be done.

Sampling will not be performed. The individuals whose professional practice will be observed will be determined and advised by the [organisation].

Recruitment procedures
How will you identify and approach potential participants?

The [organisation] have already kindly offered access.

Consent

Give details of how informed consent will be gained and attach copies of information sheet(s) and consent form(s). Give details of how participants can withdraw consent and what will happen to their data in such a case (see the Research Ethics web site for an advisory document).

The [organisation] developers have already agreed to take part (they are a team of about 20 developers). However, we (one of the supervisors and the research student) will confirm with the [organisation] developers that we may observe their practice at one of their daily meetings (the Stand-up), referring to the Information Sheet and written Consent Form. Their participation will be re-confirmed appropriately at a subsequent Stand-up if required (for example, a change in team composition). We will record the dates of such agreements at Stand-ups, including the names of all individuals involved. The Information Sheet and Consent Form will be available to all concerned, including other collaborator contacts before they agree to take part.

An individual may withdraw at any point. Any data that they have provided will then be destroyed if they so request and there will be no resultant adverse consequences.

Methodology

Outline the method(s) that will be employed to collect and analyse data.

The research student will be learning the basics of ethnographic studies. The work will be observational, recorded mainly by contemporaneous fieldwork notes and, where appropriate, photographs. The analysis will be qualitative and ethnographic.

Data Protection

Give details of registration of the project under the DP Act and the procedures to be followed re: storage and disposal of data to comply with the Act.

Helen Sharp and Hugh Robinson are registered with DPA. Data will be stored in locked cabinets and/or on secure data servers.

Recompense to participants

Normally, recompense is only given for expenses and inconvenience, otherwise it might be seen as coercion/inducement to participate. Give details of any recompense to participants.

None

Deception

Give details of the withholding of any information from participants, or misrepresentation or other deception that is an integral part of the research. Any such deception should be fully justified.

No deception

Risks

Detail any foreseen risks to participants or researchers and steps that will be taken to minimise/counter these. If the proposed study involves contact with children or other vulnerable groups, please confirm that an enhanced CRB Disclosure has been obtained for each person involved in these contacts.

Participants may be concerned that their positions are compromised by us observing their work. Participants will be allowed to withdraw at any time and we will stop collecting data if they become uncomfortable.

No risk to researchers.

Debriefing

Give details of how information will be given to participants after data collection to inform them of the purpose of their participation and the research more broadly.

A 'feedback' session will be held with the [organisation].

Declaration

Declare here that the research will conform to the above protocol and that any significant changes or new issues will be raised with the HPMEC before they are implemented.
A **Final Report** form will need to be filled in once the research has ended (you will be contacted by HPMEC on the date for final report below).

Signature(s)   Helen Sharp, Hugh Robinson, Jennifer Ferreira
(this can be the typed name(s) of investigator(s) if electronic copy is submitted (which is preferred))

Date   30.11.07

**Proposed date for
Final Report**   June 2011

Information Sheet: Interaction design and Agile Software Development in Practice

We're delighted that one of the Agile development teams at [organisation] will be helping us with our research into interaction design and Agile software development. The research is investigating how interaction design techniques are integrated with Agile software development in practice. This information sheet gives you some background on the research and what is involved in taking part in the research project.

Who will carry out the research?

The research team is from the Computing Department at the Open University and consists of a PhD student, Jennifer Ferreira, and her supervisors, Helen Sharp and Hugh Robinson. Their details appear at the end of this document. Jennifer Ferreira will be carrying out most of the work with the development team.

What is the research about?

Interaction designers and Agile developers aim to produce usable software that is valuable to the customer. Our research is about how professional software development teams integrate interaction design techniques with Agile practices. It is an area that's crucially important to development teams following Agile methods and the interaction designers they work with, but it's also an area that isn't as well understood as it should be. Our research aims to improve that understanding by sharing our findings with the community via appropriate journals and conferences.

This study builds on preliminary results from our first case study with a professional Agile team. In that study we observed 14 professionals and carried out interviews with project managers, developers and an interaction designer, to learn about how they combined interaction design techniques with their Agile practices. We will extend the previous work by comparing those findings with what we learn at [organisation], refine existing research questions and identify areas for further study.

What does the research involve?

Observational fieldwork involves a researcher observing, in a non-intrusive fashion, some aspect of the team's daily activities and capturing details via field notes and, sometimes, photographs. Informal conversations (that occur during lunch or other natural breaks in the work) between the researcher and participants may provide further insights into the team's activities.

How will I know what's happening on a day-to-day basis?

Jennifer will attend the daily standup and will keep you informed of what's planned for the day in terms of observational work, etc. and whether or not it involves you.

What if I don't want to be involved?

For whatever reason, that's fine – just let us (Jennifer, or any member of the team) know and we will respect your wishes. You can also request that any data that you have provided is destroyed. You can do this for the whole study, if you wish, or as and when you need to on a daily basis.

What about the results of the research? ... and confidentiality?

All our research activities will be conducted so as not to bring [organisation] into disrepute. After some analysis of the field notes, we will hold a feedback session to discuss our findings.

We expect to eventually publish our research findings in appropriate journals and conferences (such as the Agile and XP conferences). Any use of the research data from the development team will be anonymised so that the identity of individuals remains confidential. We will discuss with both the development team and [organisation] as to whether you wish the team and company identity to remain confidential. We will, of course, respect your wishes.

Is this research ethical?

We at the Open University are committed to high standards of professional conduct in all research activities. As researchers, we are required to adhere to guidelines provided by the university's Human Participants and Materials Ethics Committee, who approve research undertaken with human participants. One of the requirements is that participants acknowledge on a separate consent form that they understand the purposes of the research and all that participation entails.

Jennifer Ferreira
PhD Student
j.ferreira@open.ac.uk

Prof. Helen Sharp
Principal Supervisor
h.c.sharp@open.ac.uk

Prof. Hugh Robinson
Co-supervisor
h.m.robinson@open.ac.uk

Consent Form: Interaction Design and Agile Software Development in Practice

The Open University's Human Participants and Materials Ethics Committee requires that all participants in our observational study, give their written consent:

I agree to take part in this research project and I understand that I may withdraw from the study at any time. An Information Sheet highlighting the salient points about this research, including ethical and confidentiality considerations, has been made available to me.

Signed _____   Date _____

Email:_____

APPENDIX C

Sample data

The data presented in this appendix is taken from Study 1. As mentioned in chapter 4, data was collected at the sprint planning meeting, daily standups, the retrospective meeting, meetings called between the developers and UX designers during the sprint and other naturally occuring conversations in the research setting. Excerpts from the field notes that illustrate the separateness of the UX designers and Agile developers are presented in the following sections. The interview with the UX designer, conducted after the observation period, is also included.

The raw ethnographic data are a "fragmentary corpus" Crabtree et al. [2012] of observations that hold little meaning for those other than the researcher. After being immersed in and actively engaging with the setting during the course of the study, the field notes have been used by the researcher as an aide memoire, and have been refined into a coherent analytic account via the process that was detailed in chapter 3. The collection of excerpts below have been selected and grouped as key exhibits complementing the account of work in Study 1 (see chapter 4). The number of excerpts appearing in each section is not an indication of how significant that section is, rather the key points of each section together, gave rise to the themes presented in chapter 4 that explain how and why the developers and designers worked together in the ways that they did.

C.1 Unavailability of UX designers

In the developers' attempts to make contact with the UX designers there were times when the designers were not available to respond:

[**1.FN.1.4**] 2pm: There's some problem with the visual design from UX [hadn't arrived yet]. Project manager's been trying to get in touch with the interaction designer,

however can not get in contact — mobile switched off.

[**1.FN.4.3**] RN looking for a UX person (who happens to be in a meeting and unavailable) ...He says he wastes a lot of time looking for people.

[**1.FN.5.2**] RN ascends to UX to make sure the visual designs are on their way.

[**1.FN.6.9**] RN just went to see a UX person to talk some design things through ...he comes back to talk to MS about the solution he and *[UX designer]* came up with.

[**1.FN.RNint1**] RN looks for a UX person about twice a day. He'd like more interaction, especially when the design goes into the iterations. But it seems like UX are just going through the motions. When there are no designs to work on it makes it difficult to report on progress within the team.

[**1.FN.RNint3**] RN thinks he initiates most of the interaction between developers and UX people — in attempts to find definitive answers. He finds it frustrating. It all depends on who's in charge at the moment. He feels there's a schism between UX and the developers.

C.2   Developer-designer tensions

Examples from the field notes that indicate tensions between developers and designers stemmed from issues around who owns the requirements, the UX designers taking too long to pass designs in to the developers, UX designers being unaware of the developers' schedule and UX designers not communicating design decisions to the developers.

C.2.1   Who owns the requirements?

[**1.FN.2.5**] RH explaining to MS why UX's designs are always problematic: "We (CSD) think in terms of the system in a real world with constraints, because of the different systems that have to work together."

[**1.FN.1.1**] Conversation with Project manager: UX people do design ...usability concerns the non-functional aspects of the product.

[**1. FN.6.1**] Conversation with Project manager The developers that work with UX (on the UX floor) are only doing prototyping. The proper developers (Project manager's team, for e.g.) have the use of UX in a hands-off way. The proper developers make the final product.

[**1.FN.6.4**] UX want to do quality control.

[**1.FN.6.4**] The UX boss is a very powerful person in the organisation — more powerful than the one for the development division.

[**1.FN.2.4**] UX designer: According to organisation's research: "People lie when asked their age."

[**1.FN.3.1**] (UX/dev scrum @ 10) Project manager: "Is there any UX in parental guidance?" Product owner: "We need to work out a technical solution first."

[**1.FN.3.4**] Product owner owns the requirements for *[Project name1]*.

[**1.FN.3.4**] Requirements are captured as user stories which are entered on a wiki which Product owner administrates.

[**1.FN.3.8**] There was a bit of discussion among the developers about how under 16s can just enter an email address/check a check box without really having parental consent. And a comment from one of the group was that the whole thing was really meaningless and just a way for the organisation to cover themselves. UX have given no justification, other than that users lie when entering date of birth and there is the legal requirement for parental consent for under 16s.

[**1.FN.6.1**] Project manager's comment about 'proper' development.

[**1.FN.6.5**] Project manager feels the only way he knows what UX are doing is through what's fed to him at the UX scrum.

[**1.FN.6.6**] UX have apparently come back to Project manager and said that before they go live, they want to review what the developers have done.

[**1.FN.6.7**] Project manager: (The UX/dev tension) "It's about power."

[**1.FN.6.8**] T brought up a UX issue that she mentioned to Project manager and he says he'll feed back what she just said to UX. T: "There's no save button ... I'm just thinking from a practical user point of view." According to MS the Javascript version of the screen saves your details as you enter them into the fields, so there is no save button. There is a save button on the non-Javascript version. MS thinks this is an oversight on the part of UX but agrees to wait for the final designs before coming to a decision as to what to do.

[**1.FN.7.2**] Yesterday in their meeting with *[Name]* the visual design was shown by UX but they hadn't emailed it to Project manager's team. Product owner: "Cause that's all we need so we can get started."

[**1.FN.9.4**] Date of birth discussion in the retrospective.

[**1.FN.RNint1**] B and RN had a vision at the beginning for *[Project name1]*. They sketched everything out and were very passionate. They could have done it all in 6 months, but their enthusiasm wasn't tapped for this project.

[**1.FN.RNint2**] Since RN and B's sketches, UX took 12 months to come up with a "real pass" of the wireframes and visual designs.

[**1.FN.RNint2**] The requirement that *[Project name1]* needed to scale to the whole organisation site came from UX.

[**1.FN.RNint3**] Everyone needs to trust each other's motives. At the moment UX and developers are from different tribes, gong in different directions as each understand the user/product. CSD also have strong opinions about what would be best for users, but they're also thinking about making the developers' lives easier. UX don't care about technology. They have an abstract and conservative view of user needs ... a very academic approach to UX ... UX people think we don't understand the users because applying knowledge of the user to development looks different ... remind people that we're on the same side.

C.2.2   UX take too long to feed developers with designs

[**1.FN.1.1**] Conversation with Project manager: they take very long to deliver their designs ... developers are just waiting.

[**1.FN.1.3**] This project has had 12 months of UX work. Backend work is now done.

[**1.FN.1.4**] It's 2pm and the visual design still hasn't arrived.

[**1.FN.1.9**] (at 16:20 after the Sprint Planning Meeting) Visual designs arrive from UX.

[**1.FN.1.9**] Some estimation at the planning meeting could not be done because of the visual designs not appearing.

[**1.FN.5.1**] B: (10:30 scrum) "Without UX it looks shit" (he's done some work that affects the visual design, but because the visual designs have not yet arrived, he has implemented a temporary visual design).

[**1.FN.5.1**] RN: (10:30 scrum) "I'm trying to find out what's going on with that." (the visual designs not arriving).

[**1.FN.5.1**] Project manager: (10:30 scrum) "Today's the final delivery of everything." (all the designs).

[**1.FN.5.1**] (10:30 scrum) RN still has no closure about the non-Javascript parts of the wireframes/visual designs. He asked Project manager if he can get in touch with the creative director or some such person and Project manager said yes but that he was expecting the designs late today. So it seems he's telling RN to hang on and wait until the end of today.

[**1.FN.5.3**] Project manager is still hopeful that the wireframes and visual designs will arrive today. Apparently the UX person's machine crashed today and had to go home to finish the designs.

[**1.FN.6.1**] UX designer: (talking about the wireframes sent through to the developers at the end of the previous day) "Pretty final"

[**1.FN.6.1**] Senior UX designer: "On the design front there has been a computer crisis, but are confident the visual designs will be done by the end of today."

[1.FN.6.4] Product owner: (in developer scrum meeting) "We'll get the final designs later today."

[1.FN.6.5] Project manager: Creative people want to start from scratch, they don't want to take over someone else's work. This is a problem they've had many times before. Then, the time it takes the UX person to catch up with where the other UX person left off — this jeopardises the project on Project manager's side.

[1.FN.6.10] (@16:05) Final visual design has not arrived yet.

[1.FN.7.1] (@10:30 developer scrum) RN is liaising with designers and making guesses about the visual design which turned out to be acceptable: "Basically because I said this is how it's going to be."

[1.FN.7.1] (Somewhere between the end of Monday and beginning of Wednesday) wireframes have come through and UX have signed off on it.

[1.FN.7.2] Visual designs have still not come through. Yesterday, in their meeting with *[Name]* the visual design was shown by UX but they hadn't emailed it to the developers. Product owner: "Cause that's all we need so we can get started."

[1.FN.7.3] MB + MNZ + B discuss how DoB could be displayed. MB goes off and D joins in, discussion moving to how DoB is stored and how age is calculated. B insists that the solution should not be a 'hack'. They agree the whole thing is rubbish and D and MNZ get back to working together.

[1.FN.7.4] Project manager about UX: "They're not agile ... they can't make changes quickly ... they take very long ... they only pay lip-service to agile."

[1.FN.9.1] (@11:00) "Wireframes are here!" Called out by someone.

[1.FN.9.1,2] (in developer retrospective meeting @ 11:00) Good that wireframes were delivered ... but bad that they were delivered so late.

[1.FN.9.3] RN: (in developer retrospective meeting @ 11:00) "There will be more things coming but nothing that will hold us up as much as it has been."

[1.FN.RNint2] Since RN and B's sketches, UX took 12 months to come up with a "real pass" of the wireframes and visual designs — but there were other issues that influenced why UX took so long, i.e. "there were too many variables for coming up with a useful design."

C.2.3 UX work schedule does not take development schedule into account

[1.FN.2.6] (in developer feedback session) UX have a review meeting tomorrow but will aim to have missing pieces to the developers tomorrow.

[1.FN.3.9] (following on from the developer feedback session) Some, but not all, of the missing pieces come through at 15:15.

[1.FN.6.1] UX are moving desks (not all the visual designs have been delivered since the developer feedback session).

[1.FN.6.4] A new visual designer has been assigned to Project manager's team.

[1.FN.2.6] UX have a review meeting tomorrow but will aim to have the missing pieces to the developers by tomorrow (promised during the developer feedback meeting with UX (re visual designs))

[1.FN.6.1] UX designer: (talking about the wireframes sent through to the developers at the end of the previous day) "Pretty final"

[1.FN.6.1] UX designer: (talking about the wireframes sent through to the developers at the end of the previous day) "In my mind it's pretty done."

[1.FN.6.3] Project manager to Product owner " ...an analogy would be a surgeon dropping his tools halfway."

[1.FN.6.7] Project manager: The UX person being taken off the project at this stage of the development effort, is like a surgeon walking out of an operation before he's done. It wouldn't happen, but you'd be upset if you were the patient. Who's supposed to finish the job?

[1.FN.6.4] (at developer scrum meeting) Project manager tells the team the new visual designer will be starting today and they will get the final designs done.

[1.FN.6.5] The UX person has been taken off this project (since about Friday last week) and L is the new UX person (met her at today's UX scrum).

[1.FN.6.5] Project manager: Creative people want to start from scratch, they don't want to take over someone else's work. This is a problem they've had many time before. Then, the time it takes the UX person to catch up with where the other UX person left off — this jeopardises the project on Project manager's side.

[1.FN.6.9] I thought (judging from this morning's UX/dev scrum meeting) that L was the replacement UX person. Now RM mentions J.

[1.FN.6.10] Project manager: .When new UX people arrive on the project, they tend to go over old ground. Stuff which developers have already sorted out with previous UX person, they need to go over again.

[1.FN.7.1] (Somewhere between the end of Monday and beginning of Wednesday) wireframes have come through and UX have signed off on it.

[1.FN.7.2] Visual designs have still not come through. Yesterday, in their meeting with [Name] the visual design was shown by UX but they hadn't emailed it to the developers. Product owner: "Cause that's all we need so we can get started."

[1.FN.7.4] Project manager about UX: "They're not agile ...they can't make changes quickly ...they take very long ...they only pay lip-service to agile."

[1.FN.9.1] (@11:00) "Wireframes are here!" Called out by someone.

[**1.FN.9.1,2**] (in developer retrospective meeting @ 11:00) Good that wireframes were delivered ... but bad that they were delivered so late.

[**1.FN.9.3**] RN: (in developer retrospective meeting @ 11:00) "There will be more things coming but nothing that will hold us up as much as it has been."

[**1.FN.9.6**] D (UX) just came down to find RN who's left his desk to tell him that it occurred to him that having a 'save' button on one of the forms didn't make sense ...

[**1.FN.RNint2**] New UX people get added to the project and are only on for short periods.

C.2.4 Design decisions not communicated with developers

[**1.FN.2.3**] (In developer feedback meeting with UX designer) the developers want to know why aspects of the design have changed from previously agreed versions.

[**1.FN.2.6**] (In UX feedback meeting) UX promise to go away and show all details of all user journeys (for the Befit of the developers), e.g. when do spinners appear, how errors appear.

[**1.FN.2.6**] developer: (In UX feedback meeting) talking about errors popping out the side: "That would be difficult ... We don't have the time ... don't know — we'd have to experiment.

[**1.FN.3.7**] Developers had questions why the DoB behaviour as explained in the wireframe document is different to what had been given to them before.

[**1.FN.3.8**] There was a bit of discussion among the developers about how under 16s can just enter an email address/check a check box without really having parental consent. And a comment from one of the group was that the whole thing was really meaningless and just a way for the organisation to cover themselves.

[**1.FN.4.4**] Visual designs will finalise error message issue. This shows the prominence of the visual designs and how design decisions are communicated to developers via the visual designs.

[**1.FN.6.5**] Project manager feels the only way he knows what UX are doing is through what's fed to him at the UX scrum.

[**1.FN.7.2**] Yesterday in their meeting with *[Name]* the visual design was shown by UX but they hadn't emailed it to Project manager's team. Product owner: "Cause that's all we need so we can get started."

[**1.FN.8.1**] When the UX person and MS went through the wireframes, they discuss everything from a visual consistency perspective (comparing wireframes to other wireframes in the pile that are similar) not in a user journey perspective.

C.3 Gap analysis: finding the gaps

Part of transforming the UX designs into working software, involved inspecting the designs handed over by the UX designers, comparing it to the software already implemented and identifying mismatches between them:

[1.FN.1.8] (In Sprint Planning Meeting) Need to show exactly what team have/will do against UX designer's wireframes, the wireframes are a wider picture of what could be done but doesn't show what is absolutely necessary.

[1.FN.1.10] "The wireframes are not enough — we'll have to fill in the gaps in which case we may as well have started from scratch."

[1.FN.1.9 (In Sprint planning meeting) wireframes need to be translated into Javascript and in the process, gaps in the wireframes need to be identified — difference between design and implementation.

[1.FN.2.3 Project manager: (talking about undo and cancel features in the UX/dev meeting) "That's a wireframe for the future ... that's out of scope."

[1.FN.2.3] (In developer feedback meeting with UX designer) the developers want to know why aspects of the design have changed from previously agreed versions.

[1.FN.2.5] B asks RN to go through the wireframes to create cards for the wireframes that haven't got cards, and add those to the board — difference between design and work-in-progress.

[1.FN.2.6] Developers found that errors and hints were missing from one of the wireframes.

[1.FN.2.6] (In UX feedback meeting) Together they find that hints are missing from one of the wireframes? consistency? so they annotate the printed out wireframes.

[1.FN.3.9 The problem RN is trying to get closure on is that some of the visual designs didn't show how errors would appear. By 3:15 UX have solutions for some of the issues raised at yesterday's meeting with UX but not all.

[1.FN.4.2] (after RN and MS discussed the need for a '404' card) RN and MS are looking through their emails for the most recent design of a '404' page.

[1.FN.6.6] MNZ asks RN if an under 16 has parental consent (and the parent's email address has been verified), if the parent's email address changes, do they still have parental consent? Does the new address have to be verified before consent is restored? ... MNZ suggests some relabelling of the field.

[1.FN.6.8] T brought up a UX issue that she mentioned to Project manager and he says he'll feed back what she just said to UX. T: "There's no save button ... I'm just thinking from a practical user point of view." According to MS the Javascript version of the screen saves your details as you enter them into the fields, so there is no save button. There is a save button on the non-Javascript version. MS S thinks this is an oversight on the part of UX but agree to wait for the final designs before coming to a decision as to what to do.

[**1.FN.7.1**] (@10:30 developer scrum) RN is liaising with designers and making guesses about the visual design which turned out to be acceptable: "Basically because I said this is how it's going to be."

[**1.FN.7.5**] (@15:05) T comes to RN bearing printouts of the wireframes affecting her work to ask a question. Then returns to her desk.

[**1.FN.8.1**] They talk back and forth about presentation details ... the text in the dialog boxes ... need the visual design to show how UX designer's text coming out the side should look.

[**1.FN.Rnint2**] Gap analysis is just about seeing what things we need to do. It is necessary especially when something is already started, to see where the gaps are. Previous versions were approximately representations of what was in our minds, so after the fact. A third of the gap analysis leads into UX things.

C.4   Understanding UX designs and preparing feedback

Part of transforming the UX designs into working software, involved developers investing time in becoming familiar with the UX designs that were sent to them, building up an understanding of what the designs meant:

[**1.FN.1.9**] Developers crowd around RN's computer, wondering about the status if the visual designs

[**1.FN.2.2**] RN, RH + MS discuss wireframe printouts at RN's desk, going over ideas about the wireframes for discussion in the meeting with UX

[**1.FN.2.2**] RN had sent an email to the other developers and UX designer with his comments about the wireframes.

[**1.FN.2.3**] (In developer feedback meeting with UX designer) the developers want to know why aspects of the design have changed from previously agreed versions.

[**1.FN.2.3**] RN explains how undo could work [developers giving interaction design suggestions]

[**1.FN.2.4**] [discussing the appearance of UI elements ito consistency]

[**1.FN.2.4**] RN wants comic strip of how day/month/year should work [developers requesting more detail/supporting artefacts]

[**1.FN.2.6**] developers request details of all user journeys [developers requesting more detail/supporting artefacts]

[**1.FN.2.6**] developers request details of dynamic UI aspects, i.e. screen transitions and error messages [developers requesting more detail/supporting artefacts]

[**1.FN.2.7**] developer: "What if we do something, email it to them and ask 'how does that look?"'

[**1.FN.3.2**] (after 10:30 scrum) B + Project manager + RN discuss develop wireframes story.

[**1.FN.3.7**] Developers had questions why the DoB behaviour as explained in the wireframe document is different to what had been given to them before.

[**1.FN.3.9**] Some things in the wireframes/visual designs are implied and not spelled out.

1.FN.4.5 RN highlights different areas of wireframes/visual designs to help make sure things are consistent.

1.FN.6.4 (In developer scrum meeting) Get all our comments on wireframes together today for *[Name]* to get a look at tomorrow.

[**1.FN.6.8**] RH gets UX to come down and talk about breadcrumbs.

[**1.FN.7.3**] MS + MNZ + B discuss how DoB could be displayed. MS goes off and joins in, discussion moving to how DoB is stored and how age is calculated. B insists that the solution should not be a 'hack'. They agree the whole thing is rubbish and D1 and MNZ get back to working together.

[**1.FN.8.1**] MS: "I'm never sure whether wireframes are supposed to show position . . . " UX: "Yes . . . roughly."

[**1.FN.8.1**] (@10:10) UX person comes down and talks to MS about which wireframes' visual designs are needed. Many of them are the same so they go through the latest wireframes to identify which visual design would be most necessary. The UX person marks and annotates the printed out wireframes that MS tells him they need the visual designs for.

[**1.FN.9.7**] (RN to D) "There are no standards for that . . . what should we do there?"

C.5 Carding and prioritising UX design

Part of transforming the UX designs into working software, required the developers to identify the remaining implementation work according to the latest version of the UX designs, write those tasks on cards and then merge the cards with the current backlog:

[**1.FN.2.5**] B asks RN to go through the wireframes to create cards for the wireframes that haven't got cards, and add those to the board — difference between design and work-in-progress.

[**1.FN.3.2**] The cards for the wireframes/visual designs are arranged on the glass wall next to the standard wall.

[1.FN.3.2] (after 10:30 scrum) B + Project manager + RN discuss develop wireframes story.

[1.FN.4.1] (Before the 10:30 scrum meeting) B adds 2 SE tasks to the wall today under 'Develop wireframes' story

[1.FN.4.1] RN adds a pink card to 'creating wireframes' story in the 'In progress' column

[1.FN.4.1] B adds a white story card to the glass wall.

[1.FN.4.1] RH asks B if he's made any other additions on the glass wall. He did.

[1.FN.4.1] RN adds white cards to the glass wall

[1.FN.4.1] RN and MS discuss the need for a '404 page doesn't exist' card. B wants to make the card.

[1.FN.8.1] (@10:10) UX person comes down and talks to MS about which wireframes' visual designs are needed. Many of them are the same so they go through the latest wireframes to identify which visual design would be most necessary. The UX person marks and annotates the printed out wireframes that MS tells him they need the visual designs for.

C.6   Interview with the UX designer

[1.IN.20/02/09.1 12] Held on 20 February, 2009.

1. What is your background and is this typical for interaction designers at the organisation? You said you did a Masters in HCI and ergonomics from UCL.

UX designer: I think it's always difficult talking about typical background because it's a young enough field that people come from all over the place. But the UCL masters is a very popular one and very well recognised and pretty much everyone who has done the masters ends up working in user experience. I'll know one or two people at most of the agencies who's done the masters. So you're not surprised if someone's done the masters in HCI, but it's not the majority.

[Organisation-related question removed to retain anonymity.]

2. How are the different UX people assigned to development teams? And would other projects also have, like *[Project manager]*'s team, an interaction designer and visual designer?

UX designer: Yes. It depends on the size of each project and what their needs are. So there

are actually some projects without any UX resource on them at all. It's not quite an agency model but the way it works is a Product Owner will say, "I reckon I need some design work done, or some UX input," and they'll submit a brief to the UX resource manager or head of UX saying, "I think I need this. Can you help me out?" And then it'll be decided that actually you need an IA, or actually it's a quick piece of visual design, or actually it's an extended piece of work that you need a dedicated resource on. And so I might be put on a project and then I say, "Actually I really need a visual designer on this, or I need a usability person to take care of the usability testing." It's gone feral in a few cases, specifically *[Project name1]*, where there were about fourteen UX people full time on *[Project name1]* who have now kind of branched off into their own little team and actually getting quite divorced from the central UX team — because they all sit with the *[Project name1]* developers. I'm actually in that team even though the majority of what I do is *[Project name2]*. So *[Project name1]* is fourteen people permanently on it. It has a creative director who creatively directs other things as well but sits in *[Project name1]* 'cause it's the major thing. Four or five visual designers, a couple of interaction designers, IAs and a usability person. Whereas other projects have nothing, no UX resource. Like *[Project name3]*, for example, is one I'm fighting to get someone put on it because we really should. But just the way things have worked there is no UX people on it.

3. Can you describe the process that UX follow when designing a product?

UX designer: The design process is very different and depends on the type of people involved in the project. So we don't follow one process. We haven't invented a magic process that we've given a sexy name like some agencies do. *[Project name2]* is one of these projects, there's a few of them at the organisation, where no one really remembers how it started. It's back in the mists of time; it's been going on forever. But it was something to do with the fact that the current log-on system, *[Project name4]*, was inadequate. They really needed a new one and they've now got a new backend platform that is really good. So they need to rebuild the whole thing, start from scratch, so they did about two years ago. *[Project name2]* started and that's my understanding of how it happened. But each one is different.

4. You mentioned the Product Owner — are they usually developers?

UX designer: Product Owners wouldn't be in UX. UX can and does lead projects, which is a different model to what I've just described, but I don't really see it. The only example

I know of is the homepage redesign. The current homepage was started off by UX and they've ran and managed it but they did get an official Product Owner on the development team and so on from what's now called the *[Business unit name]*. Which is kind of back to front, which is great. There's no reason why UX shouldn't kick off projects and it's really successful as well.

5. When are users involved in the process and do you get to iterate your designs with users?

UX designer: User research is ad hoc. The way I find it tends to work is, my impression of the management philosophy around it, if they have a conscious one or not I don't know, but the way it tends to work out is the resources are in place to do it — space, time and budget will be given to you if you make a case for doing it. And so management make sure resources are available and they try to put the right people on the right projects and they say, "Go do your thing."

6. So UX would be making the case to do user research?

UX designer: Yes. Sometimes UX will pay for it and sometimes the project will pay for it. Depends. But it's very much driven by the UX people on the project, saying "Listen, we really need to do some research here and here's our proposal, give us some money." They would have to convince whoever's paying for it and the Product Owner that this is a good idea to do this.

7. Are UX people assigned different projects?

UX designer: I've been on more than one project for quite a long time. For the first six months I worked here I was 100% on *[Project name1]*. But I mean *[Project name1]* is a project with lots of subprojects. July I was on *[Project name2]* 100% and then from about September I was on *[Project name2]* approximately 60% of the time and 40% back on *[Project name1]*. Just today I got on another *[Project name1]* project for *[Project name5]*, which is pretty cool. So technically I'm on three projects at the minute.

8. Do you think your interaction with the developers works well? Why?

UX designer: My take on it is it's needs must. So when I was the only person on *[Project name1]*, I demanded to sit with the developers — caused a minor rumpus at the time — and I was just part of their team. I was in the *[Project name1]* team, but I wasn't. I was actually covert; I was in the UX team but for all intents and purposes I was on the *[Project*

name1] team, and I went down to the pub with them and I was involved in their sprint planning and did everything a developer did in terms of meetings, organisation, I did it too. I was just the UX guy. And I thought it was very important because during sprint planning, when they're prioritising things, I'll say, "Well actually this thing here may be technically very easy, or technically low priority, but actually it's got a very big..." you know it may be a copy change, changing a label, "And actually that's got a big UX impact, so can we bump it up the priority please?" And I'm trying to convince them to do stuff like that. And usually they'll go, "No, shut up." But sometimes they'll go "Yeah, fair enough." And they'll do it. Since *[Project name1]* blossomed ... that's a bit too positive a way of describing it, but blossomed into a large UX team, it's been a bit more divided. So the creative director on it wants it to be ... he doesn't like it being anarchic at all. He wants to know everything that's going on and therefore all design decisions go through him.

9. Is that *[Name]*?

UX designer: No. *[Name]* is head of *[Business unit name1]*. Originally *[Name]* was in charge of *[Project name1]*. He was then promoted a few months ago to Head of *[Business unit name1]*. *[Business unit name1]* is responsible for building and delivering everything that the content teams don't do, which means the homepage, *[Project name1]*, *[Project name2]*, centralised backend platforms, *[Project name6]*. He's in charge of all those now and *[Project name1]* is his big one.

10. So the person you're talking about is the creative director.

UX designer: is the creative director of the UX team. So him or his kind of right hand woman will work closely with the *[Project name1]* scrum master-type person. But the UX people themselves don't sit in on any of their planning. I've described *[Project name1]*, I haven't really described *[Project name2]*. I did start off sitting with them [*[Project name2]*]. My team leader didn't like ... he is quite purist and he thinks that if the UX people go and sit with developers they get infected with ideas about thinking that technical feasibility is an excuse for not doing the optimal user experience and heretical notions like that, so he wants the UX people to be separate, sit with themselves and exchange UX ideas and to go and convince the developers that actually this is the right thing to do and if you disagree, we'll stomp around a lot or we'll prove it or we'll get some research saying actually this is the right thing to do. And so they can be nice and pure and divorced away from technical

230

issues. Which is fine — it's a point of view — but I'm more pragmatic and I think that actually that process of getting developers and product owners and so on to buy in to your design decisions means you've got to get on well with them. It's all about politics and relationships and things like that and there's no better way to build good relationships with the product teams than to sit with them and go to the pub with them and being part of the everyday hubbub and interactions of sitting together, which is why I like to do that. So occasionally my team leader says "No, you're too weak, you're being infected. I can tell." And will pull me away. So that's why I sat with the *[Project name2]* guys for a while, but now I pulled out and I'm now sitting with the *[Project name1]* UX people who sit next to the *[Project name1]* development team and I'm just around the corner from the *[Project name2]* developers so it's actually a really happy balance. I still think I'm right though.

11. What has been the biggest challenge with respect to working with the developers and their Agile development process? While I was observing it seemed to be that developers and designers were working according to different schedules. For example, the developers didn't have the wireframes when they were planning their sprint.

UX designer: that was never explicitly said to me. So maybe they didn't even realise it themselves I guess, or they didn't think that the solution was to tell me, or whatever, perhaps this didn't even cross their minds, or they thought that I didn't want to know, or I wanted to be separate because we're strange UX people who sit on high horses and dictate what users want and so on and don't want to go down to the level of mere developers. I think possibly they were just assuming that that was the status quo and actually you couldn't really change it and it's something you have to work with, whereas actually you could. They could say to my team leader, "Listen, we really need him to sit with us and we want him to sit in our planning ... and stuff." Or they could have told me and I could have just done it anyway and not tell my team leader. I think there was and there still is visibility issues to do with what they're doing and what their timelines are. I think we're a lot better now and we're being a lot more systematic now and we have things like we have lists of items and delivery dates that are mutually agreed and things are prioritised and agreed between both. So I'll say, "Here's a list of things that are currently wrong with what you're building from the UX point of view and here's the UX priority. What's your

take on it?" and they'll go, "Oh we can't do that. And they'll send me a list with their priorities." That works pretty well. That's been tried recently and we're doing that but it's still not formally part of their sprint planning. They then take that list and they go, "Right, here's our sprint planning and here's the bit of paper that the UX designer sent us." So I'm imagining that they will just add that as an item but I don't know. I just send them an email with what I think needs fixed from the UX point of view. They're a black box really to me, I just check the website and say "That's not fixed yet, oh they've fixed that, great." I've no visibility of what their backlog is.

12. It just seems that things like that should be shared.

UX designer: well it was when I was last there, when I sat with them and I was all part of that. I think it's partly ... so from my end, why am I not doing it: It's a good question. I never really thought about it but I think it's possibly time. I see a three hour, two and a half hour meeting every two weeks on a Monday. And I think I have better things to do. And so much of it is irrelevant. 95% of it is stuff that's got no point on what I do; I have no view on what it does. I think I did go to one or two of them. So I think that was why. I realised well actually this is a waste of my time. But see the 5% of stuff that is relevant to me will help if I was there for that.

13. How did you decide when you were going to get those designs back to them, and how does the work that results from the developer feedback get put back into your schedule?

UX designer: so the way we're working at the moment is not very agile. We have a mammoth document. We found this the only way we can deal with tracking all the design decisions and the number of dialogs and variations that there are in [Project name2] the only way we can do it is by having a mammoth wireframe and user flows document. Which I was very reluctant to do and literally took months of pestering to convince me to do it because that's the least interesting part of the job for me. I have to admit it is very useful and I'm just coming to the completion of the second iteration of that document which is much nicer than the first one which was a bit of a mess. We just had a review of it yesterday. It's really good because when somebody says something like, "How do we deal with a fourteen year old who registers with their email address but their parent doesn't give explicit consent but it's on the site that requires and then validates their email before they agree to the house

rules?" Before the document it was like "Um, so do we send them an extra confirmation email? Then we needed a copy on that didn't we? Did we get the editorial person to do the copy?" "Oh yeah I forgot to do that. Allright let's make sure we do that for next week. Ok." We'll meet up next week and discuss it and then next week it's like "What were we talking about?" "I've no idea." And we'll just forget about it until it comes up as a problem again. Whereas now it's all laid out in the document. A page saying, here's how we deal with this and here's the flow for it. So here's what we're building, it's got these problems, it's not the ideal UX because of this this and this but at least we agree on it and let's do this for now. At the end of the document I've got improvements. So page 32 we're actually going to skip this part of the flow and do this or update the wireframe to do this and they'll take those and add those to the backlog. So it's just a big record really.

14. Is it just for UX or can developers also make use of it?

UX designer: this is interesting. Originally it was just to manage UX decisions, as a record of them, between UX and the *[Project name2]* team. That's what the document was done for, however *[Project name2]* is a project that's going to be used by many websites across the organisation and so a lot of them are clamouring to see how it's actually going to look like and how it's going to behave because they're either just curious or it's going to impact on their website design. So what's happened is it's gone out to other people. That's kind of ok. So at least they can see this is how it will work but ah they haven't about the seventeen year olds who are migrating from *[Project name4]* but have only one adopted parent. Then we'll go, oh actually page 17. Oh yes they have thought of it — good. And also they'll say, "Well we wanted it done this way but it's in the wireframes done that way, so oh well." And they stop arguing which is brilliant. The problem is they get the wrong idea and start building bits of it themselves. They think they have to build it, which is quite strange. So we recently had a site come back to us who wanted to do *[Project name2]* and said "Oh we've looked at the wireframes and the designs and everything and here's what we're going to do." They totally changed all our stuff — "we're going to change this dialog and change this to suit our spec." This was a totally different site, a totally different division from across the organisation. And they just stole this document and they thought it was like some kind of guide for everyone to implement *[Project name2]* in the way that they want, they thought. And they came back to us and said, "Just to let you know, here's how we're

building *[Project name2]* on our site." And we're like "Uh no, what are you talking about. You're not allowed to. *[Project name2]* is like an off-the-shelf thing. It will work like this on every site. That's the point. You're not allowed to customise it, or rescan it, or add bits."

15. At what point do you think of the UX work as 'done'?

UX designer: the fun never ends really. UX work never finishes on a product, but it will finish on subprojects of that product. So *[Project name2]* is never finished. *[Project name1]* is never finished.

16. What do you continue to do on that?

UX designer: Well the product owner will have a crazy idea about what he wants to do next and he realises he needs some UX to add some feature, or whatever, or UX run a bit of final usability testing and it turns out things are absolutely perfect, or they say, "Actually we need to fix this." And the developers go "Actually, yeah, ok." So work continues. *[Project name2]* is interesting because from the outside it's a log in dialog, why has it been going on for two years? And the UX has been going on for two years as well, what is the problem? It's just changes in scope, stakeholders changing their minds or discovering new things, realising "Oh actually we can't do that because of this arcane legal requirement we forgot about — we've got to actually change it." That happening, but to a massive degree because there are so many stakeholders that have views on it. Decisions not being made, things going in circles. At the minute it's just completing the spec. It's a large spec. Not all of it has been built yet. Building is continuing. There's enough of a base that it's live on the *[Project name1]* message board right now, but there's lots of components of it that need to be done when other sites use it. Because they need different parts of it and those different parts haven't been built yet. And as they're being built it turns out that we didn't think of this and didn't think of that and that requires UX work. Also it's me looking back at my own work and looking at what's been built and going "Ok that's actually really awful, it really should be like this." And going back and saying, "Listen, can we change that bit." And they're like "Yeah, we'll add it to the bottom of the list." It's just a log in dialog, but it's not.

17. I remember how under sixteens had to enter a full date of birth and over sixteens don't.

UX designer: yeah, that's not how it has been built. It's now mandatory for everyone —

day month and year. And no one is happy with that decision. The reason why is because it's the simplest thing to do. And it was *[Name]*, who is ultimately in charge of *[Project name2]*, who came in and said, "Why has *[Project name2]* not gone live?" That's because of date of birth. He said "Mandatory for everyone." Actually he said, "What's the simplest thing to build? I don't care about anything else, what's the simplest thing to build, that means we can go live tomorrow?" Mandatory for everyone is the simplest thing to build but we really don't want it. "Mandatory for everyone. That's the way it is. We'll change it afterwards. We'll release it and change it afterwards." He's actually quite right considering ... but it's one of these ones that it really shouldn't be mandatory for everyone for many reasons and everyone agrees. The problem isn't getting everyone to agree, it's just that there's so many people that by the time the news of what it is has gone out here, they're coming back, because what they think it is is something different to what everyone agrees it is. They come back saying, "We disagree with date of birth," and it's like "Oh, I thought everyone agreed." But they're thinking of something else. It's a big mix up and confusion and meanwhile there's another three teams — there's *[Business unti name2]*, there's *[Business unti name3]*, there's *[Business unti name4]* — and they've all got these views on what it should be. Turns out everyone's views are actually the Product ownere, but there's just different versions of it floating around. Everyone at a different time thinks it's something else. And so it's everyone just shouting and if everyone would just be quiet for a moment it would be fine. That's my take on what happened. Hopefully everyone will be quiet now and everyone will say, "Mandatory for under sixteens. Over sixteens don't have to give a date of birth. All they have to say is I'm over sixteen and that's it." Everyone is happy with that so hopefully that's what's going to be built.

18. Why do UX design both wireframes and visual designs? Is this how every development project works or was it just the *[Project name2]* project that was done that way?

UX designer: no pretty much every project has wireframes and visual designs done.

19. Is it for the document?

UX designer: so when it was me and *[Another UX designer]* on *[Project name1]* a year ago, we often skipped doing wireframes at a certain stage and we went straight to the visual design. Because it was just quicker. I'm averse to doing massive wireframe specs and it was

just that we were going at such a break-neck speed that it was just, let's skip it out and we'll talk about change, and just change the visual design. So it's not like it's always like that but currently, the way things work is the wireframes are done and the visual design is done and I think a large part of it is organisational. It's nearly like IAs' and interaction designers' job is to do a wireframe and the visual designers' job is to do the visual design. So you're never going to have a project with just a visual designer doing it. If it's about functionality or behaviour or flows or thinking about being user-centred, thinking about what are the user requirements here and how am I going to implement those in the interface. Everyone would agree that that's interaction design and the best way to specify that is through wireframes. Visual designers often focus on, even like the fact that they're able to focus on doing a great visual design and not having to get too worked up about is it drop-down list or is it a ... thingy, or what is the precise functionality here. Because they let someone else make those sorts of decisions.

20. So they're dependent on the wireframes being done so they can do the visual design?

UX designer: Yes it's a common complaint among visual designers that they don't have the wireframes and they're guessing. And then people come along saying, "That functionality's wrong." And give the visual designer a hard time. "Well I didn't have any wireframes, I had to make something up!"

21. What is [Name]'s relation to UX? Did you ever meet with him?

UX designer: Yes. He does have a say over UX. He is responsible for [Project name1] — what it is, what it looks like, how it works, when it's delivered everything. He is the one boss in charge of [Project name1]. So a while ago there was a bit of a power struggle between the head of UX and [Name] because the head of UX at the time argued that they're UX decisions and he is responsible for UX decisions and [Name] builds and delivers [Project name1] and is responsible for that part of it. The way that it was resolved was that [Name] won. That head of UX has actually moved to a different job. [Name] is responsible for everything. UX, organisationally, does not report to [Name] and that's definitely a cause of conflict. Technically, the creative director of [Project name1], who's responsible for delivering the user experience of [Project name1], does not report to the head of [Project name1] directly. So I don't report to [Name] at all but [Name] is in charge of [Project name2] ultimately

so he will overrule my decisions. Technically what I should do is I escalate it to head of UX and I'll say "Listen *[Name]*'s wrong and you need to go and have a fight with him." No one wants to have a fight with *[Name]* because they all know they'll lose. So actually the only way to do it is to convince *[Name]* and be rational and say "I think you actually made the wrong decision here and here's why." And hope he reverses his decision. He does sometimes and sometimes he doesn't but that's it — he's the boss so that's that.

REFERENCES

P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta. *Agile Software Development Methods: Review and Analysis.* VTT Publications 478, Espoo, 2002.

P. Abrahamsson, J. Warsta, M. Siponen, and J. Ronkainen. New directions on Agile methods: A comparative analysis. In *ICSE '03: Proceedings of the 25th International Conference on Software Engineering, Portland, Oregon, USA, May 3–10*, pages 244–254, Washington, DC, USA, 2003. IEEE Computer Society.

P. Abrahamsson, A. Hanhineva, H. Hulkko, I. Tuomas, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen, and O. Salo. Mobile-D: An agile approach for mobile application development. In *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, OOPSLA '04, pages 174–175, New York, NY, USA, 2004. ACM. ISBN 1-58113-833-4. doi: http://doi.acm.org/10.1145/1028664.1028736. URL http://doi.acm.org/10.1145/1028664.1028736.

P. Abrahamsson, M. Babar, and P. Kruchten. Agility and architecture: Can they coexist? *Software, IEEE*, 27(2):16 –22, March–April 2010. ISSN 0740-7459. doi: 10.1109/MS.2010.36.

S. Adikari, C. McDonald, and J. Campbell. Little design up-front: A design science approach to integrating usability into agile requirements engineering. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part I: New Trends*, pages 549–558, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02573-0. doi: http://dx.doi.org/10.1007/978-3-642-02574-7_62. URL http://dx.doi.org/10.1007/978-3-642-02574-7_62.

M. H. Agar. *The Professional Stranger: An Informal Introduction to Ethnography.* Academic Press Inc., London, UK, 2nd edition, 1980.

M. Albisetti. Launchpad's quest for a better and agile user interface. In W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, A. Sillitti, A. Martin, X. Wang, and E. Whitworth, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 48 of *Lecture Notes in Business Information Processing*, pages 244–250.

Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-13054-0. URL http://dx.doi.org/
10.1007/978-3-642-13054-0_26. 10.1007/978-3-642-13054-0_26.

S. W. Ambler. Tailoring usability into Agile software development projects. In E. Law,
E. Hvannberg, and G. Cockton, editors, *Maturing Usability*, Human-Computer Interaction Series, pages 75–95. Springer, London, UK, 2008.

J. Armitage. Are Agile methods good for design? *interactions*, 11(1):14–23, 2004. ISSN
1072-5520. doi: http://doi.acm.org/10.1145/962342.962352.

P. Baheti, E. Gehringer, and P. Stotts. Exploring the efficacy of distributed pair programming. In *Proceedings of the Second XP Universe and First Agile Universe Conference on
Extreme Programming and Agile Methods*, LNCS 2418, pages 208–220. Springer, August
2002.

R. S. Barbour and M. Barbour. Evaluating and synthesizing qualitative research: The
need to develop a distinctive approach. *Journal of Evaluation in Clinical Practice*, 9(2):
179–186, May 2003.

J. Barksdale, E. Ragan, and D. McCrickard. Easing team politics in agile usability: A
concept mapping approach. In *Agile Conference, 2009. AGILE '09*, pages 19–25, Aug
2009. doi: 10.1109/AGILE.2009.57.

K. Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Reading,
MA, USA, 1999.

K. Beck and C. Andres. *Extreme Programming Explained: Embrace Change*. Addison-
Wesley, Boston, MA, USA, 2nd edition, 2004.

K. Beck and M. Fowler. *Planning Extreme Programming*. Addison-Wesley, Boston, MA,
2001.

J. Belenguer, J. Parra, T. I., and P. J. Molina. HCI designers and engineers: Is it possible
to work together? In M. B. Harning and J. Vanderdonckt, editors, *CLOSING THE
GAPS: Software Engineering and Human-Computer Interaction*, pages 14–19. Université

catholique de Louvain, Institut d'Administration et de Gestion (IAG), Louvain-la-Neuve, Belgium, 2003.

H. Beyer. *User-Centered Agile Methods*. Synthesis Lectures on Human-Centered Informatics. Morgan & Claypool, www.morganclaypool.com, 2010.

H. Beyer and K. Holtzblatt. *Contextual Design: Defining Customer-Centered Systems.* Morgan Kaufmann Publishers Inc., San Francisco, CA, 1997.

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 '04: Extreme Programming and Agile Methods, Calgary, Alberta, Canada, August 15–18*, pages 50–59. vol. 3134 of Lecture Notes in Computer Science, Springer, 2004.

F. O. Bjørnson and T. Dingsøyr. Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Inf. Softw. Technol.*, 50:1055–1068, October 2008. ISSN 0950-5849. doi: 10.1016/j.infsof.2008.03.006. URL http://portal.acm.org/citation.cfm?id=1405197.1405310.

S. Blomkvist. Towards a model for bridging Agile development and user-centered design. In A. Seffah, J. Gulliksen, and M. C. Desmarais, editors, *Human-centered software engineering: Integrating usability in the development process*, pages 219–244. Springer, Dordrecht, Netherlands, 2005.

B. Boehm and R. Turner. Observations on balancing discipline and agility. In *Proceedings of the Conference on Agile Development*, pages 32–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-2013-8. URL http://portal.acm.org/citation.cfm?id=942789.942828.

B. Boehm and R. Turner. Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22:30–39, 2005. ISSN 0740-7459. doi: http://doi.ieeecomputersociety.org/10.1109/MS.2005.129.

V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006. doi: 10.1191/1478088706qp063oa. URL http://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa.

D. Broschinsky and L. Baker. Using persona with XP at LANDesk Software, an Avocent company. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 543–548, Los Alamitos, CA, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3321-6. doi: http://doi.ieeecomputersociety.org/10.1109/{A}gile.2008.91.

J. Brown, G. Lindgaard, and R. Biddle. Stories, sketches, and lists: Developers and interaction designers interacting through artefacts. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 39–50, Los Alamitos, CA, USA, 2008. IEEE Computer Society. doi: 10.1109/{A}gile.2008.54.

M. Budwig, S. Jeong, and K. Kelkar. When user experience met agile: A case study. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI '09, pages 3075–3084, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4. doi: http://doi.acm.org/10.1145/1520340.1520434. URL http://doi.acm.org/10.1145/1520340.1520434.

B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design.* Morgan Kaufmann Publishers, 2007.

J. M. Carroll. *Scenario-Based Design.* Wiley, New York, NY, USA, 1995.

J. Carver, L. Jaccheri, S. Morasca, and F. Shull. Issues in using students in empirical studies in software engineering education. In *Software Metrics, IEEE International Symposium on*, page 239, Los Alamitos, CA, USA, 2003. IEEE Computer Society. ISBN 0-7695-1987-3. doi: http://doi.ieeecomputersociety.org/10.1109/METRIC.2003.1232471.

S. Chamberlain, H. Sharp, and N. A. M. Maiden. Towards a framework for integrating Agile development and user-centred design. In *XP '06: Proceedings of the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering, Oulu, Finland, June 17–22*, pages 143–153, 2006.

L. Cho. Adopting an agile culture. In *Agile Conference, 2009. AGILE '09*, pages 19–25, Los Alamitos, CA, USA, Aug 2009. IEEE Computer Society. doi: http://doi.ieeecomputersociety.org/10.1109/AGILE.2009.76.

T. Chow and D.-B. Cao. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6):961–971, June 2008.

M.-W. Chung and B. Drummond. Agile at Yahoo! from the trenches. In *Agile Conference, 2009. AGILE '09*, pages 113–118, Aug. 2009. doi: 10.1109/AGILE.2009.41.

T. Coatta and J. Gosper. UX design and agile: A natural fit? *Queue*, 8(11), 2010. doi: 10.1145/1874534.1891739.

A. Cockburn. Characterizing people as non-linear, first-order components in software development. Technical Report 1999.03, Humans and Technology, Oct 21 1999. Available at `http://alistair.cockburn.us/Characterizing+people+as+non-linear,+first-order+components+in+software+development`. Last accessed 21 July, 2011.

A. Cockburn. *Agile software development*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002. ISBN 0-201-69969-9.

A. Cockburn. *Crystal Clear: A Human-Powered Methodology for Small Teams*. Addison-Wesley Professional, 2004.

A. Cockburn. *Agile Software Development: The Cooperative Game*. Addison-Wesley, 2nd edition, 2006.

A. Cockburn and J. Highsmith. Agile software development, the people factor. *Computer*, 34(11):131–133, Nov 2001. ISSN 0018-9162. doi: 10.1109/2.963450.

A. A. R. Cockburn. The impact of object-orientation on application development. *IBM Syst. J.*, 32(3):420–444, 1993. ISSN 0018-8670.

L. Constantine, R. Biddle, and J. Noble. Usage-centered design and software engineering: Models for integration. In *Proceedings of the workshop Bridging the Gaps Between Software Engineering and Human-Computer Interaction held at ICSE'03: International Conference on Software Engineering, Portland, OR, May 3-11*, pages 106–113, 2003.

L. L. Constantine. Process agility and software usability: Toward lightweight usage-centered design. *Information Age*, 8(8):1–10, 2002.

L. L. Constantine and L. A. D. Lockwood. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. ACM Press, 1999.

L. L. Constantine and L. A. D. Lockwood. Usage-centered software engineering: An agile approach to integrating users, user interfaces, and usability into software engineering practice. In *Proceedings of the 25th International Conference on Software Engineering*, ICSE '03, pages 746–747, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1877-X. URL http://portal.acm.org/citation.cfm?id=776816.776931.

A. Cooper. *The Inmates are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. SAMS, 1999.

A. Cooper. Common ground. Available at http://www.cooper.com/journal/2010/03/common_ground.html. Last accessed 11 June, 2011, March 2010.

A. Cooper and R. Reimann. *About Face 2.0: The Essentials of Interaction Design*. Wiley Publishing, Indianapolis, IN, 2003.

A. Cooper, R. Reimann, and D. Cronin. *About Face 3.0: The Essentials of Interaction Design*. Wiley Publishing, Indianapolis, IN, 2007.

J. Coutaz and R. N. Taylor. Introduction to the workshop on software engineering and human-computer interaction: Joint research issues. In *Software Engineering and Human-Computer Interaction, ICSE'94 Workshop on SE-HCI: Joint Research Issues, Sorrento, Italy, May. 1994*, pages 1–3, Berlin, Heidelberg, 1995. Springer-Verlag. ISBN 3-540-59008-0.

A. Crabtree, M. Rouncefield, and P. Tolmie. Analysing the ethnographic record. In *Doing Design Ethnography*, HumanComputer Interaction Series, pages 111–133. Springer London, 2012. ISBN 978-1-4471-2726-0. URL http://dx.doi.org/10.1007/978-1-4471-2726-0_7. 10.1007/978-1-4471-2726-0_7.

T. S. da Silva, A. Martin, F. Maurer, and M. Silveira. User-centered design and agile methods: A systematic review, 2011. *Accepted to the Agile 2011 Conference, Salt Lake City, UT, August 7–13*.

D. Dayton and C. Barnum. The impact of agile on user-centered design: Two surveys tell the story. *Technical Communication*, 53(3):219–234, August 2009.

E. Derby, D. Larsen, and K. Schwaber. *Agile Retrospectives: Making Good Teams Great*. Pragmatic Bookshelf, 2006.

M. Detweiler. Managing UCD within Agile projects. *interactions*, 14(3):40–42, 2007. ISSN 1072-5520. doi: http://doi.acm.org/10.1145/1242421.1242447.

M. Düchting, D. Zimmermann, and K. Nebe. Incorporating user centered requirement engineering into Agile software development. In *Human-Computer Interaction. Interaction Design and Usability*, Lecture Notes in Computer Science, pages 58–67. Springer, Berlin/Heidelberg, 2007.

T. Dybå and T. Dingsøyr. Empirical studies of agile software development: A systematic review. *Inf. Softw. Technol.*, 50:833–859, August 2008. ISSN 0950-5849. doi: 10.1016/j.infsof.2008.01.006. URL http://portal.acm.org/citation.cfm?id=1379905.1379989.

S. Faraj and L. Sproull. Coordinating expertise in software development teams. *Management Science*, 46(12):1554–1568, December 2000.

M. Federoff and C. Courage. Successful user experience in an agile enterprise environment. In *Proceedings of the Symposium on Human Interface 2009 on Conference Universal Access in Human-Computer Interaction. Part I: Held as Part of HCI International 2009*, pages 233–242, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02555-6. doi: http://dx.doi.org/10.1007/978-3-642-02556-3_27. URL http://dx.doi.org/10.1007/978-3-642-02556-3_27.

M. Federoff, C. Villamor, L. Miller, J. Patton, A. Rosenstein, K. Baxter, and K. Kelkar. Extreme usability: Adapting research approaches for Agile development. In *CHI '08: CHI '08 extended abstracts on human factors in computing systems*, pages 2269–2272, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358666.

X. Ferre. Integration of usability techniques into the software development process. In *Proceedings of the ICSE Workshop on Bridging the Gaps between Software Engineering and HumanComputer Interaction*, 2003.

X. Ferre, N. Juristo, and A. Moreno. Framework for integrating usability practices into the software process. In F. Bomarius and S. Komi-Sirviö, editors, *Product Focused Software Process Improvement*, volume 3547 of *Lecture Notes in Computer Science*, pages 202–215. Springer, Berlin / Heidelberg, 2005. URL http://dx.doi.org/10.1007/11497455_17. 10.1007/11497455_17.

J. Ferreira. Interaction design and Agile development: A real-world perspective. Master's thesis, Victoria University of Wellington, New Zealand, 2007.

J. Ferreira. Interaction design and Agile development: Reconciling iterative and incremental approaches. Work in progress presented at Agile '08: The Agile 2008 Conference, Toronto, Ontario, Canada, August 4 8, 2008.

J. Ferreira, J. Noble, and R. Biddle. Agile development iterations and ui design. In *Agile '07: Proceedings of the Agile 2007 Conference, Washington, DC, USA, August 13-17*, pages 50 58, Los Alamitos, CA, USA, 2007a. IEEE Computer Society. ISBN 0-7695-2872-4. doi: http://dx.doi.org/10.1109/AGILE.2007.8.

J. Ferreira, J. Noble, and R. Biddle. Up-front interaction design in Agile development. In G. Concas, E. Damiani, M. Scotto, and G. Succi, editors, *Agile Processes in Software Engineering and Extreme Programming*, pages 9–16. Lecture Notes in Computer Science, Springer, Berlin / Heidelberg, 2007b. ISBN 978-3-540-73100-9. doi: 10.1007/978-3-540-73101-6_2.

J. Ferreira, J. Noble, and R. Biddle. Interaction designers on extreme programming teams: Case studies from the real world. In *NZCSRSC '07: Proceedings of the 5th New Zealand Computer Science Research Student Conference*, 2007c.

J. Ferreira, H. Sharp, and H. Robinson. Values and assumptions shaping agile development and user experience design in practice. In W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, A. Sillitti, A. Martin, X. Wang, and E. Whitworth, editors, *Agile*

Processes in Software Engineering and Extreme Programming, volume 48 of *Lecture Notes in Business Information Processing*, pages 178–183. Springer, Berlin Heidelberg, 2010. ISBN 978-3-642-13054-0. URL http://dx.doi.org/10.1007/978-3-642-13054-0_15. 10.1007/978-3-642-13054-0_15.

J. Ferreira, H. Sharp, and H. Robinson. User experience design and agile development: Managing cooperation through articulation work. *Software: Practice and Experience*, 41 (9):963–974, 2011. ISSN 1097-024X. doi: 10.1002/spe.1012. URL http://dx.doi.org/10.1002/spe.1012.

J. Ferreira, H. Sharp, and H. Robinson. User experience design and agile development integration as an on-going achievement in practice. In *Agile '12: Upcoming Proceedings of the Agile 2012 Conference, Dallas, TX, USA, August 13–17*, Los Alamitos, CA, USA, 2012. IEEE Computer Society.

K. Fisher and A. Bankston. From cradle to sprint: Creating a full-lifecycle request pipeline at nationwide insurance. In *Agile Conference, 2009. AGILE '09*, pages 223–228, Aug 2009. doi: 10.1109/AGILE.2009.72.

E. Folmer, M. v. Welie, and J. Bosch. Bridging patterns: An approach to bridge gaps between SE and HCI. *Inf. Softw. Technol.*, 48:69–89, February 2006. ISSN 0950-5849. doi: http://dx.doi.org/10.1016/j.infsof.2005.02.005. URL http://dx.doi.org/10.1016/j.infsof.2005.02.005.

J. Forlizzi and K. Battarbee. Understanding experience in interactive systems. In *Proceedings of the 5th conference on Designing interactive systems: Processes, practices, methods, and techniques*, DIS '04, pages 261–268, New York, NY, USA, 2004. ACM. ISBN 1-58113-787-7. doi: http://doi.acm.org/10.1145/1013115.1013152. URL http://doi.acm.org/10.1145/1013115.1013152.

M. Fowler. Is Design Dead?, 2004. Published online at: http://martinfowler.com/articles/designDead.html#id20933. Updated May 2004. Last accessed 11 June, 2011.

M. Fowler. The New Methodology. Published online at http://www.martinfowler.com/

`articles/newMethodology.html`. Updated December 2005. Last accessed 11 June, 2011, 2005.

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 '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 63–72, Los Alamitos, CA, USA, 2008. IEEE Computer Society. doi: 10.1109/{A}gile.2008.54.

A. Frank and C. Hartel. Feature teams collaboratively building products from ready to done. In *Agile Conference, 2009. AGILE '09*, pages 320–325, Aug 2009. doi: 10.1109/ AGILE.2009.51.

J. J. Garrett. *The Elements of User Experience: User-Centered Design for the Web*. New Riders Publishers, Indianapolis, IN, USA, 2002.

E. Gerson and S. Star. Analyzing due process in the workplace. *ACM Trans. Inf. Syst.*, 4 (3):257–270, 1986.

B. Göransson, J. Gulliksen, and I. Boivie. The usability design process — integrating user-centered systems design in the software development process. *Software Process: Improvement and Practice*, 8:111–131, 2003. doi: 10.1002/spip.174.

T. L. Greenbaum. *Moderating Focus Groups: A Practical Guide for Group Facilitation*. Sage Publications Inc., 2000. ISBN 0761920447.

J. Grudin and R. T. Fielding. Working group on design methods and processes. In *Proceedings of the Workshop on Software Engineering and Human-Computer Interaction*, ICSE '94, pages 4–8, London, UK, 1995. Springer-Verlag. ISBN 3-540-59008-0. URL `http://portal.acm.org/citation.cfm?id=645541.657989`.

J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and A. Cajander. Key principles for user-centred systems design. *In the special section on Designing IT for Healthy Work of the International Journal Behaviour and Information Technology*, 22 (6):397–409, 2003.

J. Gulliksen, A. Cajander, and E. Eriksson. Only figures matter? — if measuring usability and user experience in practice is insanity or a necessity. In *Proceedings of the International Workshop on Meaningful Measures: Valid Useful User Experience Measurement, Reykjavik, Iceland, 18 June*, pages 91–96, 2008. Available at http://141.115.28.2/cost294/upload/523.pdf#page=93.

M. Haesen, K. Coninx, J. Bergh, and K. Luyten. Muicser: A process framework for multi-disciplinary user-centred software engineering processes. In *Proceedings of the 2nd Conference on Human-Centered Software Engineering and 7th International Workshop on Task Models and Diagrams*, HCSE-TAMODIA '08, pages 150–165, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-85991-8. doi: http://dx.doi.org/10.1007/978-3-540-85992-5_14. URL http://dx.doi.org/10.1007/978-3-540-85992-5_14.

J. Haikara. Usability in Agile software development: Extending the interaction design process with personas approach. In G. Concas, E. Damiani, M. Scotto, and G. Succi, editors, *Agile Processes in Software Engineering and Extreme Programming*, Lecture Notes in Computer Science, pages 153–156. Lecture Notes in Computer Science, Springer, Berlin / Heidelberg, 2007.

J. Hakim, T. Spitzer, and J. Armitage. Sprint: Agile specifications in Shockwave and Flash. In *DUX '03: Proceedings of the 2003 conference on Designing for user experiences, San Francisco, California, USA*, pages 1–14, New York, NY, USA, 2003. ACM. ISBN 1-58113-728-1. doi: http://doi.acm.org/10.1145/997078.997111.

M. Hassenzahl and N. Tractinsky. User experience — a research agenda. *Behaviour and Information Technology*, 25(2):91–97, March–April 2006. doi: 10.1080/01449290500330331.

J. Highsmith. *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House Publishing Co., Inc., 2000.

J. Highsmith. *Agile Software Development Ecosystems*. Pearson Education Inc., Boston, MA, USA, 2002.

J. Highsmith. *Agile Project Management: Creating Innovative Products*. Addison Wesley Professional, 2009.

J. Highsmith and A. Cockburn. Agile software development: The business of innovation. *Computer*, 34(9):120–127, Sep 2001. ISSN 0018-9162. doi: 10.1109/2.947100.

J. Higman, T. Mackinnon, I. Moore, and D. Pierce. Innovation and sustainability with gold cards. In *Extreme Programming Perspectives*. Addison-Wesley Professional, Indianapolis, IN, 2002.

R. Hoda. *Self-Organizing Agile Teams: A Grounded Theory*. PhD thesis, Victoria University of Wellington, New Zealand, 2011.

P. Hodgetts. Experiences integrating sophisticated user experience design practices into Agile processes. In *Agile '05: Proceedings of the 2005 Agile Conference, Denver, CO, USA, July 24–29*, pages 235–242, Los Alamitos, CA, USA, 2005. IEEE Computer Society. doi: 10.1109/ADC.2005.24.

A. Holzinger and W. Slany. XP + UE − > XU Praktische Erfahrungen mit eXtreme Usability. *Informatik Spektrum*, 29(2):91–97, 2006. ISSN 0170-6012.

K. L. Hope and E. Amdahl. Configuring designers? using one agile project management methodology to achieve user participation. *New Technology, Work and Employment*, 26 (1):54–67, March 2011.

A. Hosseini-Khayat, Y. Ghanam, S. Park, and F. Maurer. Activestory enhanced: Low-fidelity prototyping and wizard of oz usability testing tool. In W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, P. Abrahamsson, M. Marchesi, and F. Maurer, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 31 of *Lecture Notes in Business Information Processing*, pages 257–258. Springer, Berlin Heidelberg, 2009. ISBN 978-3-642-01853-4. URL http://dx.doi.org/10.1007/978-3-642-01853-4_57. 10.1007/978-3-642-01853-4_57.

W. Hudson. Adopting user-centered design within an agile process: A conversation. *Cutter IT Journal*, October 2003. Available at URL: http://www.syntagm.co.uk/design/articles/ucd-xp03.pdf.

Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, and P. Wolkerstorfer. Agile user-centered design applied to a mobile multimedia streaming application.

In A. Holzinger, editor, *HCI and Usability for Education and Work*, volume 5298 of *Lecture Notes in Computer Science*, pages 313–330. Springer, Berlin / Heidelberg, 2008a. ISBN 978-3-540-89349-3. URL http://dx.doi.org/10.1007/978-3-540-89350-9_22. 10.1007/978-3-540-89350-9_22.

Z. Hussain, M. Lechner, H. Milchrahm, S. Shahzad, W. Slany, M. Umgeher, and P. Wolkerstorfer. Integrating Extreme Programming and User-Centered Design. In *PPIG '08: Proceedings of the 20th annual meeting of the Psychology of Programming Interest Group, Lancaster, UK*, pages 107–113. Lancaster University, 2008b.

Z. Hussain, H. Milchrahm, S. Shahzad, W. Slany, M. Tscheligi, and P. Wolkerstorfer. Integration of extreme programming and user-centered design: Lessons learned. In *Agile Processes in Software Engineering and Extreme Programming*, Lecture Notes in Business Information Processing, pages 174–179, 2009a. doi: 10.1007/978-3-642-01853-4_23.

Z. Hussain, W. Slany, and A. Holzinger. Investigating agile user-centered design in practice: A grounded theory perspective. In *Proceedings of the 5th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for e-Inclusion*, USAB '09, pages 279–289, Berlin, Heidelberg, 2009b. Springer-Verlag. ISBN 978-3-642-10307-0. doi: http://dx.doi.org/10.1007/978-3-642-10308-7_19. URL http://dx.doi.org/10.1007/978-3-642-10308-7_19.

Z. Hussain, W. Slany, and A. Holzinger. Current state of agile user-centered design: A survey. In *Proceedings of the 5th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for e-Inclusion*, USAB '09, pages 416–427, Berlin, Heidelberg, 2009c. Springer-Verlag. ISBN 978-3-642-10307-0. doi: http://dx.doi.org/10.1007/978-3-642-10308-7_30. URL http://dx.doi.org/10.1007/978-3-642-10308-7_30.

T. Illmensee and A. Muff. 5 users every friday: A case study in applied research. In *Agile Conference, 2009. AGILE '09*, pages 404–409, 2009. doi: http://10.1109/AGILE.2009.45. URL http://10.1109/AGILE.2009.45.

T. Jokela and P. Abrahamsson. Usability assessment of an extreme programming project: Close co-operation with the customer does not equal to good usability. In *Product Fo-*

cused Software Process Improvement, Lecture Notes in Computer Science, pages 393–407. Springer, Berlin/Heidelberg, 2004.

R. Juárez-Ramírez, G. Licea, I. Barriba, V. Izquierdo, and A. Ángeles. Engineering the development process for user interfaces: Toward improving usability of mobile applications. In *Proceedings of the International Conference on Digital Information and Communication Technology and its Applications, Part II, Dijon, France, June*, Communications in Computer and Information Science. Springer, Heidelberg, Dordrecht, London, New York, 2011. ISBN 978-3-642-22026-5.

B. Junker. Some Suggestions for the Design of Field Work Learning Experiences. In E. Hughes, B. H. Junker, R. L. Gold, and D. Kittel, editors, *Cases on Field Work*. The University of Chicago, 1952.

D. Kane. Finding a place for discount usability engineering in Agile development: Throwing down the gauntlet. In *ADC '03: Proceedings of the 2003 Agile Development Conference, Salt Lake City, UT, USA, June 25–28*, pages 40–46, Los Alamitos, CA, USA, 2003. IEEE Computer Society. doi: 10.1109/ADC.2003.1231451.

F. Keenan, N. Damdul, S. Kelly, and D. Connolly. Summary reporting for a linked interaction design-scrum approach: How much modeling is useful? In W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, P. Abrahamsson, M. Marchesi, and F. Maurer, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 31 of *Lecture Notes in Business Information Processing*, pages 245–246. Springer, Berlin Heidelberg, 2009. ISBN 978-3-642-01853-4. URL http://dx.doi.org/10.1007/978-3-642-01853-4_51. 10.1007/978-3-642-01853-4_51.

B. Kitchenham. Procedures for undertaking systematic reviews. Technical Report TR/SE-0401, Department of Computer Science, Keele University and National ICT, Australia Ltd, 2004. Joint Technical Report.

J. Kollmann, H. Sharp, and A. Blandford. The importance of identity and vision to user experience designers on agile projects. In *Proceedings of the 2009 AGILE Conference, Chicago, IL, USA, August 24–28*, pages 11–18. IEEE Computer Society, 2009. ISBN 978-0-7695-3768-9. doi: http://doi.ieeecomputersociety.org/10.1109/AGILE.2009.58.

T. Krohn, M. C. Kindsmüller, and M. Herczeg. User-centered design meets feature-driven development: An integrating approach for developing the web application mypim. In *Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International 2009*, HCD 09, pages 739–748, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02805-2. doi: http://dx.doi.org/10.1007/978-3-642-02806-9_86. URL http://dx.doi.org/10.1007/978-3-642-02806-9_86.

P. Kruchten. Agility with the RUP. *Cutter IT Journal*, 14(12):27–33, 2001.

E. L.-C. Law. Bridging the HCI-SE gaps: Historical and epistemological perspectives. In M. Harning and J. Vanderdonckt, editors, *Proceedings of the INTERACT 2003 Workshop (IFIP WG2.7/13.4): Closing the Gap between Human-Computer Interaction and Software Engineering, Zürich, September, 2003*, pages 47–54, 2003.

J. Lee, D. Scott McCrickard, and K. Stevens. Examining the foundations of agile usability with extreme scenario-based design. In *Agile Conference, 2009. AGILE '09*, pages 3–10, Aug 2009. doi: 10.1109/AGILE.2009.30.

J. C. Lee. Embracing Agile development of usable software systems. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1767–1770, New York, NY, USA, 2006. ACM. ISBN 1-59593-298-4. doi: http://doi.acm.org/10.1145/1125451.1125784.

J. C. Lee and S. McCrickard. Towards extreme(ly) usable software: Exploring tensions between usability and Agile software development. In *Agile '07: Proceedings of the Agile 2007 Conference, Washington, DC, USA, August 13–17*, pages 59–71, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

T. Lethbridge, S. Sim, and J. Singer. Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10:311–341, 2005.

M. A. Lievesley and J. S. R. Yee. The role of the interaction designer in an Agile software development process. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1025–1030, New York, NY, USA, 2006. ACM. ISBN 1-59593-298-4. doi: http://doi.acm.org/10.1145/1125451.1125647.

A. Light. Adding method to meaning: A technique for exploring peoples' experience with technology. *Behaviour and Information Technology*, 25(2):175–187, March–April 2006. doi: 10.1080/01449290500331172.

Y. Lincoln and E. Guba. *Naturalistic Inquiry*. Sage Publications, Beverly Hills, CA, 1985.

H. Lindström and M. Malmsten. User-centred design and agile development: Rebuilding the swedish national union catalogue. *The Code{4}Lib Journal*, 5:12–15, 2008. ISSN 1940-5758.

L. Lindstrom and R. Jeffries. Extreme Programming and Agile Software Development Methodologies. *Information Systems Management*, 21(3):41–52, 2004.

M. Lindvall, V. R. Basili, B. W. Boehm, P. Costa, K. Dangle, F. Shull, R. Tesoriero, L. A. Williams, and M. V. Zelkowitz. Empirical findings in agile methods. In *Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods — XP/Agile Universe 2002*, pages 197–207, London, UK, 2002. Springer-Verlag. ISBN 3-540-44024-0. URL http://portal.acm.org/citation.cfm?id=647276.722332.

T. Little. Context-adaptive agility: Managing complexity and uncertainty. *IEEE Software*, 22(3):28–35, 2005.

M. Lukanuski, M. Milano, J. de Bruin, M. Rochford, and R. Bosman. Agile or awkward: Surviving and flourishing in an agile/scrum project. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 2253–2256, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-8. doi: http://doi.acm.org/10.1145/1358628.1358662. URL http://doi.acm.org/10.1145/1358628.1358662.

T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Comput. Surv.*, 26:87–119, March 1994. ISSN 0360-0300. doi: http://doi.acm.org/10.1145/174666.174668. URL http://doi.acm.org/10.1145/174666.174668.

A. Martin. *The Role of Customers in Extreme Programming Projects*. PhD thesis, Victoria University of Wellington, New Zealand, 2009.

A. Martin, J. Noble, and R. Biddle. Experience on the human side of agile. In W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L. Morgan, and X. Wang, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 9 of *Lecture Notes in Business Information Processing*, pages 234–235. Springer, Berlin Heidelberg, 2008. ISBN 978-3-540-68255-4. URL http://dx.doi.org/10.1007/978-3-540-68255-4_35. 10.1007/978-3-540-68255-4_35.

J. Mason. *Qualitative researching*. Sage, London, UK, 2nd edition, 2002.

F. Maurer. Agile methods and interaction design: Friend or foe? In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '09, pages 209–210, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-600-7. doi: http://doi.acm.org/10.1145/1570433.1570435. URL http://doi.acm.org/10.1145/1570433.1570435.

J. McCarthy and P. Wright. *Technology as experience*. The MIT Press, Cambridge, Massachusetts / London, England, 2004. ISBN 0-262-13447-0.

P. McInerney and F. Maurer. UCD in Agile projects: Dream team or odd couple? *Interactions*, 12(6):19–23, 2005.

S. M. McMenamin and J. Palmer. *Essential Systems Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1984.

T. Memmel, F. Gundelsweiler, and H. Reiterer. Agile human-centered software engineering. In *HCI '07: Proceedings of 21st BCS HCI Group conference, University of Lancaster, UK, September*, pages 167–175. British Computer Society, 2007a.

T. Memmel, F. Gundelsweiler, and H. Reiterer. Cruiser: A cross-discipline user interface and software engineering lifecycle. In J. A. Jacko, editor, *Human-Computer Interaction. Interaction Design and Usability*, Lecture Notes in Computer Science, pages 174–183. Springer, Berlin/Heidelberg, 2007b.

T. Memmel, H. Reiterer, and A. Holzinger. Agile methods and visual specification in software development: A chance to ensure universal access. In *Proceedings of the 4th*

international conference on Universal access in human computer interaction: Coping with diversity, UAHCI'07, pages 453–462, Berlin, Heidelberg, 2007c. Springer-Verlag. ISBN 978-3-540-73278-5. URL `http://portal.acm.org/citation.cfm?id=1766311.1766364`.

T. Memmel, C. Bock, and H. Reiterer. Model-driven prototyping for corporate software specification. In J. Gulliksen, M. B. Harning, P. Palanque, G. C. Veer, and J. Wesson, editors, *Engineering Interactive Systems*, pages 158–174. Springer-Verlag, Berlin, Heidelberg, 2008. ISBN 978-3-540-92697-9. doi: http://dx.doi.org/10.1007/978-3-540-92698-6_10. URL `http://dx.doi.org/10.1007/978-3-540-92698-6_10`.

G. Meszaros and J. Aston. Adding usability testing to an agile project. In *Proceedings of the conference on AGILE 2006*, pages 289–294, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2562-8. doi: 10.1109/AGILE.2006.5. URL `http://portal.acm.org/citation.cfm?id=1155439.1155483`.

L. Miller. Case study of customer input for a successful product. In *Agile '05: Proceedings of the 2005 Agile Conference, Denver, CO, USA, July 24–29*, pages 225–234, Los Alamitos, CA, USA, 2005. IEEE Computer Society. doi: 10.1109/ADC.2005.16.

L. Miller and D. Sy. Agile user experience sig. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI EA '09, pages 2751–2754, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4. doi: http://doi.acm.org/10.1145/1520340.1520398. URL `http://doi.acm.org/10.1145/1520340.1520398`.

M. Najafi and L. Toyoshiba. Two case studies of user experience design and Agile development. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4 8*, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

A. Narasimhadevara, T. Radhakrishnan, B. Leung, and R. Jayakumar. On designing a usable interactive system to support transplant nursing. *J. of Biomedical Informatics*, 41:137–151, February 2008. ISSN 1532-0464. doi: 10.1016/j.jbi.2007.03.006. URL `http://portal.acm.org/citation.cfm?id=1332127.1332162`.

E. Nelson. Extreme programming vs. Interaction Design. Available from http://web.archive.org/web/20030621112434/http://www.fawcette.com/interviews/beck_cooper/default.asp. Last accessed 11 Dec 2009, 2002.

S. Nerur and V. Balijepally. Theoretical reflections on agile development methodologies. *Commun. ACM*, 50:79–83, March 2007. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/1226736.1226739. URL http://doi.acm.org/10.1145/1226736.1226739.

J. Nielsen. Usability engineering at a discount. In G. Salvendy and M. Smith, editors, *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, pages 394–401. Elsevier Science Publishers, Amsterdam, 1989.

J. Nielsen. Usability engineering iifecycle. *IEEE Computer*, 25(3):12–22, 1992.

G. W. Noblit and R. D. Hare. *Meta-ethnography: Synthesizing qualitative studies*. Sage Publications Inc., Newbury Park, CA, 1988.

R. Nord and J. Tomayko. Software architecture-centric methods and agile development. *Software. IEEE*, 23(2):47–53, March April 2006. ISSN 0740-7459. doi: 10.1109/MS.2006.54.

D. A. Norman and S. W. Draper, editors. *User-Centered Design*. Lawrence Erlbaum Associates Inc., Hillsdale, N.J., 1986.

H. Obendorf and M. Finck. Scenario-based usability engineering techniques in Agile development processes. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2159–2166, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358649.

J. Ohlhauser. Experience report: Design to delivery in 7 weeks. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 549–554, Los Alamitos, CA, USA, 2008. IEEE Computer Society. doi: http://doi.ieeecomputersociety.org/10.1109/{A}gile.2008.90.

V. Paelke and K. Nebe. Integrating Agile methods for mixed reality design space exploration. In *DIS '08: Proceedings of the 7th ACM conference on Designing interactive*

systems, pages 240–249, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-002-9. doi: http://doi.acm.org/10.1145/1394445.1394471.

S. R. Palmer and M. Felsing. *A Practical Guide to Feature-Driven Development.* Prentice-Hall, Upper Saddle River, NJ, 2002.

D. Parsons, R. Lal, and H. Ryu. Software development methodologies, agile development and usability engineering. In *ACIS2007: Proceedings of the 18th Australasian Conference on Information Systems 5–7 Dec 2007, Toowoomba, Australia.* The University of Southern Queensland, 2007. ISBN 9780909756963. Available at `http://www.acis2007.usq.edu.au/papers_bynum.html`. Last accessed 14 February, 2011.

F. Paternò. *Model-Based Design and Evaluation of Interactive Application.* Springer Verlag, `http://giove.cnuce.cnr.it/~fabio/mbde.html`, 1999. ISBN 1-85233-155-0.

F. Paternò. Towards a UML for interactive systems. In M. Little and L. Nigay, editors, *Engineering for Human-Computer Interaction*, volume 2254 of *Lecture Notes in Computer Science*, pages 7–18. Springer, Berlin / Heidelberg, 2001. URL `http://dx.doi.org/10.1007/3-540-45348-2_4`. 10.1007/3-540-45348-2_4.

J. Patton. Hitting the target: Adding interaction design to Agile software development. In *OOPSLA '02: OOPSLA 2002 Practitioners Reports*, pages 1–7, New York, NY, USA, 2002a. ACM. ISBN 1-58113-471-1. doi: http://doi.acm.org/10.1145/604251.604255.

J. Patton. Designing requirements: Incorporating usage-centered design into an agile sw development process. In D. Wells and L. Williams, editors, *Extreme Programming and Agile Methods. XP/Agile Universe 2002*, volume 2418 of *Lecture Notes in Computer Science*, pages 95–102. Springer, Berlin / Heidelberg, 2002b. doi: http://dx.doi.org/10.1007/3-540-45672-4_1.

J. Patton. Improving on agility: Adding usage-centered design to a typical Agile software development environment. In *ForUse2003: Proceedings of the Second International Conference on Usage-Centered Design, Portsmouth, NH, USA, October 18–22*, 2003.

J. Patton. Interaction design meets agility: Practicing usage centered design on agile development projects. In C. Zannier, H. Erdogmus, and L. Lindstrom, editors, *Extreme*

Programming and Agile Methods - XP/Agile Universe 2004, volume 3134 of *Lecture Notes in Computer Science*, pages 35–64. Springer, Berlin / Heidelberg, 2004. URL http://dx.doi.org/10.1007/978-3-540-27777-4_48. 10.1007/978-3-540-27777-4_48.

J. Patton. Finding the forest in the trees. In *Companion to the 20th annual ACM SIG-PLAN conference on Object-oriented programming, systems, languages, and applications*, OOPSLA '05, pages 266–274, New York, NY, USA, 2005. ACM. ISBN 1-59593-193-7. doi: http://doi.acm.org/10.1145/1094855.1094961. URL http://doi.acm.org/10.1145/1094855.1094961.

A. Qumer and B. Henderson-Sellers. Comparative evaluation of xp and scrum using the 4d analytical tool (4-dat). In *Proceedings of the European and Mediterranean Conference on Information Systems 2006 (EMCIS2006)*, 2006.

D. Raithatha. Making the whole product agile: A product owners perspective. In *Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming*, XP'07, pages 184–187, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-73100-9. URL http://portal.acm.org/citation.cfm?id=1768961.1769003.

P. Rannikko. User-centered design in agile software development. Master's thesis, University of Tampere, Finland, 2011.

R. Rico, M. Sánchez-Manzanares, F. Gil, and C. Gibson. Team implicit coordination processes: A team knowledge-based approach. *Academy of Management Review*, 33(1): 163–184, 2008.

L. Rising. The product champion. *STQE www.stqemagazine.com*, pages 44–48, 2003.

H. Robinson, J. Segal, and H. Sharp. Ethnographically-informed empirical studies of software practice. *Inf. Softw. Technol.*, 49(6):540–551, 2007. ISSN 0950-5849. doi: http://dx.doi.org/10.1016/j.infsof.2007.02.007.

C. Robson. *Real World Research: A resource for social scientists and practitioner-researchers*. Blackwell Publishers, Oxford, UK, 2nd edition, 2002.

R. Rorty. *Philosophy and the mirror of nature.* Princeton University Press, Princeton, NJ, 1979.

R. Rorty. *Contingency, irony, and solidarity.* Cambridge University Press, Cambridge, UK, 1989.

J. Rumbaugh, I. Jacobson, and E. G. Booch. *The Unified Modeling Language Reference Manual.* Addison-Wesley, Reading, MA, 1999.

D. Saffer. *Designing for interaction: Creating innovative applications and devices.* New Riders, Berkeley, CA, 2nd edition, 2010.

M. Sandelowski, C. I. Voils, and G. Knafl. On quantitizing. *J Mix Methods Res.*, 3(3): 208–222, July 1 2009. doi: 10.1177/1558689809334210.

E. H. Schein. Organizational culture. *American Psychologist*, 45(2):109–119, Feb. 1990. doi: 10.1037/0003-066X.45.2.109.

E. H. Schein. *Organizational Culture and Leadership.* Jossey-Bass, San Francisco, CA, 4th edition, 2010.

K. Schmidt. The organization of cooperative work: Beyond the "leviathan" conception of the organization of cooperative work. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 101–112, New York, NY, October 1994a. ACM.

K. Schmidt. Modes and mechanisms of interaction in cooperative work. Technical report, Risø National Laboratory, P.O. Box 49, DK-4000 Roskilde, Denmark, 1994b.

H. Schulze, H. Brau, S. Haasis, M. Weyrich, and T. Rhatje. Human-centered design of engineering applications: Success factors from a case study in the automotive industry. *Human Factors and Ergonomics in Manufacturing*, 15(4):421–443, 2005.

K. Schwaber. Scrum development process. In D. Patel, C. Casanave, G. Hollowell, and J. Miller, editors, *Business Object Design and Implementation: OOPSLA '95 Workshop Proceedings.* Springer-Verlag, 1995.

K. Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004.

K. Schwaber and M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002.

A. Seffah and E. Metzker. The obstacles and myths of usability and software engineering. *Commun. ACM*, 47:71–76, December 2004. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/1035134.1035136. URL http://doi.acm.org/10.1145/1035134.1035136.

A. Seffah, J. Gulliksen, and M. C. Desmarais, editors. volume 8 of *HumanComputer Interaction Series*. Springer, Dordrecht, Netherlands, 2005.

H. Sharp and H. Robinson. An ethnographic study of XP practice. *Empirical Software Engineering*, 9(4):353–375, December 2004.

H. Sharp, H. M. Robinson, and J. A. Segal. eXtreme Programming and User-Centred Design: Friend or foe? *HCI2004 Design for Life*, 2, 2004.

H. Sharp, R. Biddle, P. Gray, L. Miller, and J. Patton. Agile development: Opportunity or fad? In *CHI '06 extended abstracts on Human factors in computing systems*, CHI EA '06, pages 32–35, New York, NY, USA, 2006. ACM. ISBN 1-59593-298-4. doi: http://doi.acm.org/10.1145/1125451.1125461. URL http://doi.acm.org/10.1145/1125451.1125461.

H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, 2nd edition, 2007.

M. Singh. U-SCRUM: An Agile methodology for promoting usability. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 555–560, Los Alamitos, CA, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3321-6. doi: http://doi.ieeecomputersociety.org/10.1109/{A}gile.2008.33.

D. Sjoberg, B. Anda, E. Arisholm, T. Dybå, M. Jorgensen, A. Karahasanovic, E. Koren, and M. Vokac. Conducting realistic experiments in software engineering. In *Proceedings of the 2002 International Symposium on Empirical Software Engineering, Nara, Japan, October 3–4*, pages 17–26, 2002. doi: 10.1109/ISESE.2002.1166921.

J. K. Smith. Interpretive inquiry: A practical and moral activity. *Theory into Practice*, 31 (2):pp. 100–106, 1992. ISSN 00405841. URL http://www.jstor.org/stable/1476395.

O. Sohaib and K. Khan. Integrating usability engineering and agile software development: A literature review. In *Computer Design and Applications (ICCDA), 2010 International Conference on*, volume 2, pages 32–38, Jun 2010. doi: 10.1109/ICCDA.2010.5540916.

O. Sohaib and K. Khan. Incorporating discount usability in extreme programming. *International Journal of Software Engineering and Its Applications*, 5(1), January 2011. Available at http://www.sersc.org/journals/IJSEIA/vol5_no1_2011/4.pdf.

J. Stapleton. *Dynamic Systems Development Method — the method in practice*. Addison Wesley, 1997.

A. Strauss. The articulation of project work: An organizational process. *The Sociological Quarterly*, 29(2):163–178, 1988.

J. Sutherland. Agile principles and values. MSDN library, 2010. Available at http://msdn.microsoft.com/en-us/library/dd997578.aspx. Last accessed 5 August, 2011.

D. Sy. Adapting usability investigations for Agile user-centered design. *Journal of Usability Studies*, 2(3):112–132, May 2007.

D. Sy and L. Miller. Optimizing Agile user-centered design. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3897–3900, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358951.

S. Teasley and S. Roschelle. Constructing a joint problem space: The computer as a tool for sharing knowledge. In S. Lajoie and S. Derry, editors, *Computers as cognitive tools*, Hillsdale, 1993. LEA.

K. Tzanidou and J. Ferreira. Design and development in the "agile room": Trialing scrum at a digital agency. In *Agile Processes in Software Engineering and Extreme Programming*, Lecture Notes in Business Information Processing, pages 372–378, 2010. doi: 10.1007/978-3-642-13054-0_40.

J. Ungar. The design studio: Interface design for Agile teams. In *Agile '08: Proceedings of the Agile 2008 Conference, Toronto, Ontario, Canada, August 4–8*, pages 519–524, Los Alamitos, CA, USA, 2008. IEEE Computer Society. doi: 10.1109/{A}gile.2008.51.

J. Ungar and J. White. Agile user centered design: Enter the design studio - a case study. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2167–2178, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358650.

D. Vaughan. *The Challenger Launch Decision: Risky technology, culture and deviance at NASA*. The University of Chicago Press, Chicago and London, 1996.

D. Vaughan. Signals and interpretive work: The role of culture in a theory of practical action. In K. A. Cerulo, editor, *Culture in Mind: Toward a Sociology of Culture and Cognition*, pages 28–54. Routledge, New York, 2002.

E. Whitworth and R. Biddle. The social nature of agile teams. In *AGILE 2007*, pages 26–36, Aug 2007. doi: 10.1109/AGILE.2007.60.

E. Wilcox, S. Nusser, J. Schoudt, J. Cerruti, and H. Badenes. Agile development meets strategic design in the enterprise. In G. Concas, E. Damiani, M. Scotto, and G. Succi, editors, *Agile Processes in Software Engineering and Extreme Programming*, Lecture Notes in Computer Science, pages 208–212. Lecture Notes in Computer Science, Springer, Berlin / Heidelberg, 2007.

H. Williams and A. Ferguson. The UCD perspective: Before and after agile. In *AGILE 2007*, pages 285–290, Aug 2007. doi: 10.1109/AGILE.2007.61.

G. B. Wills, N. Abbas, R. Chandrasekharan, R. M. Crowder, L. Gilbert, Y. M. Howard, D. E. Millard, S. C. Wong, and R. J. Walters. An agile hypertext design methodology. In *Proceedings of the eighteenth conference on Hypertext and hypermedia*, HT '07, pages 181–184, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-820-6. doi: http://doi.acm.org/10.1145/1286240.1286295. URL http://doi.acm.org/10.1145/1286240.1286295.

P. Wolkerstorfer, M. Tscheligi, R. Sefelin, H. Milchrahm, Z. Hussain, M. Lechner, and S. Shahzad. Probing an Agile usability process. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 2151–2158, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358648.