Factors for successful Agile collaboration between UX designers and software developers in a complex organisation

A thesis submitted to the University of Chester
for the degree of
Doctor of Philosophy in Computer Science

By

Alexander James Jones

May 2019

Abstract

User Centred Design (UCD) and Agile Software Development (ASD) processes have been two extremely successful methods for software development in recent years. However, both have been repeatedly described as frequently putting contradictory demands on people working with the respective processes. The current research addresses this point by focussing on the crucial relationship between a User Experience (UX) designer and a software developer. In-depth interviews, an online survey, a contextual inquiry and a diary study are described from a sample of over 100 designers, developers and their stakeholders (managers) in a large media organisation exploring factors for success in Agile development cycles.

The findings from the survey show that organisational separation is challenge for agile collaboration between the two roles and while designers and developers have similar levels of (moderately positive) satisfaction with Agile processes, there are differences between the two roles. While developers are happier with the wider teamwork but want more access to and close collaboration with designers, particularly in an environment set up for Agile practices, the designers' concern was the quality of the wider teamwork. The respondent's comments also identified that the two roles saw a close – and ideally co-located – cooperation as essential for improving communication, reducing inefficiencies, and avoiding bad products being released. These results reflected the findings from the in-depth interviews with stakeholders. In particular, it was perceived that co-located pairing helped understanding different role-dependent demands and skills, increased efficiency of prototyping and implementing changes, and enabling localised decision-making. However, organisational processes, the setup of work-environment, and managerial traditions meant that this close collaboration and localised decision-making was often not possible to maintain over extended periods.

Despite this, the studies conducted between pairs of designers and developers, found that successful collaboration between designers and developers can be found in a complex organisational setting. From the analysis of the empirical studies, six contributing factors emerged that support this. These factors are 1) Close proximity, 2) Early and frequent communication, 3) Shared ideation and problem solving, 4) Crossover of knowledge and skills, 5) Co-creation and prototyping and 6) Making joint decisions. These factors are crucially determined and empowered by the support from the organisational setting and

teams where practitioners work. Specifically, by overcoming key challenges to enable integration between UCD and ASD and thus encouraging close collaboration between UX designers and software developers, these challenges are: 1) Organisational structure and team culture, 2) Location and environmental setup and 3) Decision-making. These challenges along with the six factors that enable successful Agile collaboration between designers and developers provide the main contributions of this research. These contributions can be applied within large complex organisations by adopting the suggested 'Paired Collaboration Manifesto' to improve the integration between UCD and ASD. Beyond this, more empirical studies can take place, further extending improvements to the collaborative practices between the design and development roles and their surrounding teams.

Acknowledgements

I would like to thank everyone who has provided me with support, advice and patience during the journey through my PhD. In particular, I would like to thank my supervisors Professor Nick Avis & Dr. John Kerins for all of their expertise and advice during the course of my research. Not only have they provided fantastic guidance throughout my doctoral studies, but they have also given me motivation and plenty of humour during some of the more challenging times.

I would also like to thank Deborah Wyatt, Professor Elizabeth Mason-Whitehead, Professor Tim Wheeler, Dr. June Keeling, Professor Nigel W. John and John Scott for their sound advice and support during various stages of my research studies at the University of Chester.

Special thanks also goes to my colleagues Paul Sissons, Colin Burns, Jane Murrison, Paul Crowley, Andrew Nott & Gemma Newell for their considerable time, patience and understanding during the course of my studies.

My deepest gratitude goes to Dr. Volker Thoma for his advice and support during my studies. Especially in reviewing various paper and conference submissions that have helped to shape my research.

Lastly, I would like to thanks my friends, family and above all my wife Megan, for her love, support and incredible patience – thank you for always being there.

Declaration

The material being presented for examination is my own work and has not been submitted for an award of this or another HEI except in minor particulars which are explicitly noted in the body of the thesis. Where research pertaining to the thesis was undertaken collaboratively, the nature and extent of my individual contribution has been made explicit.

Date:

Signed:

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

This research aims to bridge the gap between User Experience (UX) designers and software developers working within an Agile Software Development environment in large organisations. The focus of this thesis is concerned with the two fields of Human-Computer Interaction (HCI) and Software Engineering (SE). There is on-going debate among academics and practitioners about how Agile Software development (ASD) relates to User Centred Design (UCD), and how they are combined or integrated with one another. UCD and ASD development have roots in different disciplines, bringing with them different perspectives of software development. This presents unique challenges for designers and developers working together. While the focus of the growing body of research and literature is on the process of integration between UCD and ASD and the recommended techniques, the day-to-day collaboration methods between UX designers and software developers, remains largely unexplored.

The focus of this investigation is therefore on designers and developers working relationship in integrating UCD and ASD. Salah et al. (2014) found in their review that the work-dynamic and relationship between a designer and a developer changes in an ASD setting. This is true in particular for designers, as their job requires them to be "on call" and supply ad-hoc solutions, reviews, and feedback in a team-oriented design process such as ASD. The importance on "working software" as the main yardstick for the design and development progress is a challenge for the designer-developer relationship. This provided the main question to be explored in this research:

How do people integrate UCD with ASD in practice and what is the impact on collaboration between designers and developers in an organisational setting?

To answer this question a mixed method research approach was applied. In particular, this study explored whether designers and developers have different perspectives in terms of how ASD works in their organisation, how well the wider team (including other roles, e.g. a business analyst) work together, and whether there are specific issues in designer-developer

collaboration and communication that impact on the current processes, their goals or objectives. To investigate this an online survey was conducted with 109 respondents following initial qualitative interviews with 9 stakeholders from various teams across a large organisation. Following the analysis of this and the online survey, the research question was refined to ask:

What does 'success' look like between pairs of designers and developers working in an UCD and ASD based environment? What are barriers and what are consequences of successful paired work?

To answer this question, a qualitative approach was selected for two studies. Pairing Study A captured insights from one pair over the course of 1 week and Pairing Study B captured insights using a Contextual Inquiry with 6 design and development pairs who also participated in a Diary Study over the course of 6 weeks to gain further understanding.

As a result of the analyses across the empirical studies, the findings show that the organisation and setting has a significant impact, where the team structure and circumstances can act as a barrier for an enabler of close collaboration. However, the results gained from the pairing studies show that successful collaboration between designers and developers can be found in an organisational setting. In particular, by creating a team structure that is multi-disciplinary and embeds both roles where location isn't a barrier to frequent communication and knowledge sharing and decisions can be made at a localised level.

The next section in this chapter provides a background to the topics and terminology in this thesis (1.1). This is following by the rationale for carrying out the research (1.2). Next, the research questions are detailed (1.3) followed finally by describing the remaining thesis structure (1.4).

1.1 Background, topics and terminology

This section provides a background to the topics and terminology relevant for discussion within this thesis. This begins with background on the overall field concerned with this research and is followed by the features of User Experience Design and Agile Software Development that are relevant for this discussion.

1.1.1 Overview

Broadly, the field of Human Computer Interaction (HCI) researches the design and usage of computer technology, focusing on the interaction between people and computer systems. The term HCI was popularised in 1983 in the book 'The Psychology of Human-Computer Interaction' and the term was first used by the authors in 1980 (Card, Moran, & Newell, 1980).

Software Engineering (Oettinger, 1967) is the systematic application of engineering to the design, development, operation and maintenance of software. Anthony Oettinger first coined the term and the discipline first emerged to address poor quality software. The aim was to ensure that software was built systematically, measurably, on time, on budget, within specification and with suitable rigour. As Law & Lárusdóttir (2015) reports, At the turn of the millennium two new approaches evolved into HCI and SE which significantly changed the way we design and build software – These were User Experience Design (UX) and Agile software development (Agile) respectively.

Agile is a philosophy that was put to paper in 2001 (Beck, Beedle, Bennekum, et al., 2001). It emerged because of the need for a lightweight set of software development methods to address drawbacks of heavyweight document-driven software development methodologies— such as Waterfall (Szalvay, 2004). Similarly, User Experience was coined as a term because of limitations presented by usability and human interface design methods – UX aimed to address a wider scope, covering the design of a person's entire experience and interaction with a system (Bevan, 2008; Tullis & Albert, 2013). UX, in accordance with ISO 9241-210:2010 (para.2.15 "user experience") is subsumed by User-Centred Design (UCD) (para 4.6 "the design addresses the whole user experience") (Law & Lárusdóttir, 2015).

UCD places the user and their requirements at the centre of the design process, aiming to involve the users in a meaningful and appropriate way throughout a system's development (Gould & Lewis, 1983). Researchers and practitioners in UCD have developed methodologies, techniques and processes to achieve a positive subjective experience and high objective performance with the interface and to enable design teams to create, prototypes and test solutions before developers (programmers) are involved. UCD has been found to positively impact the results of the design and development efforts by reducing customer complaints, training needs, and increasing uptake of resulting products (Bias & Mayhew, 2005).

Agile promotes adaptive planning to counter the perceived shortcomings of traditional plan-driven methodologies, and encourages delivery of early versions of solutions that get continuously improved. This way the team is thought to be able to respond to changing user requirements or business needs. Agile teams strive to deliver an early and fast production of working code and make frequent and incremental changes. This is often achieved with highly collaborative work in short iterative cycles, with contingent user feedback. Importantly, it requires a high degree of collaboration and shared decision-making (Drury-Grogan, O'Dwyer, 2013).

Both UCD and Agile have their own specific sets of principles, practices and tools. Some of which are shared (e.g., upholding the goal of delivering user value)—and some are unique (e.g., time-boxed constraint as sprints in Scrum; limiting the amount of work in progress in Kanban; addressing the whole user experience in UCD), and some may be even incompatible (e.g., holistic design of UCD versus reductionist slicing of work in Agile and Lean).

1.1.2 UX and UCD

For the purpose of this research, it is important to clarify the term 'User Experience Design' and how it is used as part of this thesis alongside other terms for design such as 'User-Centred Design' and 'Interaction Design'.

The term 'User Experience Design' (UX) encompasses traditional human–computer interaction (HCI) design, and as (Hassenzahl & Tractinsky, 2006) argue it extends it by addressing all aspects of a product or service as perceived by its users. The term represents a variety of meanings (Forlizzi & Battarbee, 2004) and helps to escape the narrow view that

designers of the user experience focus only on the usability effectiveness (Light, 2006). UX has its roots in human factors and ergonomics, a field that, since the 1940s has had its focus on the interaction between human users, machines, and the contextual environments to design systems that address the users experience. The term 'user experience design' remains vague and can be associated with a wide variety of meanings – "ranging from traditional usability to beauty, hedonic, affective or experiential aspects of technology use" (Hassenzahl & Tractinsky, 2006). Never the less, the term has been adopted by the HCI community, and for the purpose of the discussions within this thesis it is used to refer to the specific role of using methods, tools, techniques etc. to focus on the end user within software development. Alben (2002) defined UX as "all the aspects of how people use an interactive product: the way it feels in their hands, how well they understand how it works, how they feel about it while they're using it, how well it serves their purposes and how well it fits into the entire context in which they are using [it]".

'User Centred Design' (UCD) on the other hand is a set of techniques, methods, procedures and processes and a philosophy which aims to involve the users in a meaningful and appropriate way throughout a system's development (Gould & Lewis, 1983)**.** Gould et al first proposed three principles of UCD in the mid-1980s and during the 20 odd years since then, different techniques have developed which successfully involve the user within the design process. These two terms, 'user experience design' (UX) and 'user centred design' (UCD) are often used interchangeably but there is an important distinction between the two. UX design is a discipline i.e. *what* someone does, whereas UCD can be seen as the process of *how* someone does it. The specifics of both vary and precise definitions can be elusive. UCD is seen as the dominant design process that is applied within UX and by UX designers.

The standards definition (ISO 9241-210: 2010) defines the user experience as: "A person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service." The standard describes 6 key principles that aims to ensure that the experience is user-centred (Travis, 2011):

The design is based upon an explicit understanding of users, tasks and environments - It is important to understand a user's context in their natural setting. This can be gained by observations or interviews to find out what they want, what they need, and what their environment is like.

Users are involved throughout design and development - Ensure design teams involve users in all design phases: not by running a focus group at the start of design or by providing a survey at the end. A user's involvement needs to be 'active' (e.g., not just showing designs to but engaging them in the design process too).

The design is driven and refined by user-centred evaluation - Formative and summative user-based evaluations (e.g., Usability testing) should be carried out throughout the design process to inform and refine the product or experience.

The process is iterative - The most appropriate design for a user, typically, can not be achieved first time. And it is extremely difficult, if not impossible, to capture what a user wants from a system without iteration; ideally using feedback to validate or improve the design.

The design addresses the whole user experience - It is important to consider the overarching, holistic vision of the system under development, especially the inter-dependencies among its components and the relationship between those and other contextual or emotional factors.

The design team includes multidisciplinary skills and perspectives - A siloed design team is the wrong way to approach user centred design. You need to include a range of views, including the voices of users and that of different specialisms and disciplines (e.g., accessibility specialists, domain experts, marketing, tech support, technical writers and business analysts).

All of these principles of UCD aim to achieve the production of a usable and desirable product or experience for the user. In theory, the application of these principles is well suited, subject to the integration with Agile methods which is discussed in section 1.2.2.

UX design terminology for this thesis

There is an existence of many definitions and terms for design-related disciplines and it is not necessarily useful to identify a single definition that captures everything. The literature review (Chapter 2) continues to discuss the complexities surrounding design-related

terminology in the context of the combination with Agile development. For the purpose of the rest of the discussion the following terms will be used:

design: This term is used as a collective term in this thesis to refer to various approaches that appear within the literature, without labelling these approaches to something that was not intended by the authors of the literature. Reference is made to any specific design approaches e.g. User Centred Design (UCD).

designer or UX designer: Where these terms appear in this thesis, it is a reference to a designer whose discipline is a UX designer and is involved in UX design as described in this section.

UX design: The term 'UX Design' is used as an umbrella term for this thesis, and encompasses various *design disciplines and approaches that are combined with Agile Software Development methods as discussed in this section.

1.1.3 Agile Software Development

This section introduces a background on Agile Software Development (ASD) and its relevance for this thesis.

The ASD process is where the requirements (e.g. the users' or functional requirements) are addressed by applying solutions iteratively through collaboration between self-organising, cross-functional teams (Dybå & Dingsøyr, 2008). ASD promotes adaptive planning to counter the perceived shortcomings of traditional plan-driven methodologies such as waterfall, and encourages delivery of early versions of solutions that get continuously improved. This way the team is thought to be able to respond to changing user requirements or business needs. ASD teams strive to deliver an early and fast production of working code and make frequent and incremental changes. This is often achieved through paired programming in short iterative cycles, with contingent user feedback. Importantly, it requires a high degree of collaboration and shared decision-making (Drury-Grogan & O'Dwyer, 2013).

Incremental software development methods like this trace back to 1957 (Larman & Basili, 2003) and in 1974, E. A. Edmonds wrote a paper that introduced an adaptive software

development process (Edmonds, 1974). So-called *lightweight* software development methods evolved in the mid-1990s to counter against *heavyweight* methods, which were characterised by their critics as a heavily regulated, regimented, micromanaged model of development. Some of these early lightweight methods included Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development (DSDM) (1995). All of these methods are now typically referred to as Agile methodologies, since the publication of the *Agile Manifesto* in 2001 (Beck, Beedle, Bennekum, et al., 2001).

The Agile Manifesto

In February 2001, 17 software developers got together in Snowbird, Utah to discuss lightweight, iterative software development methods. They published the "Manifesto for Agile Software Development". The values of The Agile Manifesto[1] read as follows (Beck, Beedle, Bennekum, et al., 2001):

> "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."

Twelve principles also underlie the Agile Manifesto and its values. These principles are fulfilled using different Agile methods and can be used variably to suit the needs of a project or product. Attention to these principles is important as following them alongside trying to integrate principles of UCD is a unique but important challenge. The Manifesto describes the following principles. (Beck, Beedle, & Bennekum, 2001):

- Customer satisfaction by rapid delivery of useful software
- Welcome changing requirements, even late in development
- Working software is delivered frequently (weeks rather than months)

[1] The Agile Manifesto: http://Agilemanifesto.org/

- Working software is the principal measure of progress
- Sustainable development, able to maintain a constant pace
- Close, daily co-operation between business people and developers
- Face-to-face conversation is the best form of communication (co-location)
- Projects are built around motivated individuals, who should be trusted
- Continuous attention to technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular adaptation to changing circumstances

Individuals and interactions over processes and tools – The first value in the Agile Manifesto values people rather than using processes and tools and is seen as a key factor in the success of software development project (Blomkvist, 2005). In particular this involves cohesive teams that involve individuals who are self-organising, proactive and effectively use face-to-face communication in daily work interactions.

Working software over comprehensive documentation – The second Agile value means that the focus should always be on releasing tested working software iterations which includes sufficient documentation as opposed to producing heavy weight up to date documentation.

Customer collaboration over contract negotiation – Agile values close collaboration and collocation where possible between customers and development teams. Involving close-interactions in striving to achieve customer satisfaction as opposed to relying solely on contract negotiation.

Responding to change over following a plan – The last Agile value in the manifesto welcomes changing requirements and coping with volatile requirements by discarding heavy, upfront detailed plans. This is achieved by providing the development team with the autonomy and authority to adjust and generate flexible plans that can accommodate change or alter direction.

Agile software development vs. plan-driven methodologies

One of the first methodologies to first dominate software development projects is called "waterfall". Winston Royce (Royce, 1970) coined the term in 1970 to describe a serial method for managing software projects through five stages; *Requirements, Design, Implementation, Verification and Maintenance*. Whilst this method was widely adopted and helped companies to reduce the failure rate of software development projects, 70% of projects that used this methodology still failed to meet their objectives (Szalvay, 2004). These failures caused organisations to increase the emphasis of the planning and design phases of the waterfall process, requiring more a detailed and exhaustive amount of documentation. One of the most important differences between the Agile and waterfall approaches is that waterfall features distinct stages with key deliverables (e.g. Design) whilst Agile methods have iterations rather than phases.

As detailed by the Agile Manifesto, the emphasis is on people, working software and responding to change. Agile methods are designed to deal with the "change, speed and uncertainty" (Sharp & Robinson, 2004) that commonly are difficult during 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:

> "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, p. 27).

Agile development values changes in requirements rather than restricting changes from the customer because of early commitments that mean sticking to plans drawn up at the outset of the development project (Beck, 1999).

The impact of Agile Software Development

Software processes are an important factor in software engineering and they strongly influence the outcome of a product (Higgins & Green, 2011). Therefore, a large number of software engineering processes have evolved in recent decades and the biggest change in software processes was the introduction of Agile processes as a contrast to the plan-based processes (Dahlem, 2014). The significance of ASD is that processes should be as

lightweight as possible and at its core the focus should be on frequent iterations (Ab, 2015). The output of each iteration in an Agile environment results in working code. This can be used to evaluate and respond to changing and evolving user requirements.

Over the past decade adoption of ASD has grown to the point where it is now commonplace within Software Engineering. This has to a large degree happened due to the inherent complexity of software development e.g. having a lack of repeatability and predictability. This makes rigid processes like waterfall unable to provide the adaptability that is required in software development (Ab, 2015). The number of ASD methods has grown over the years to the point where there are now 20 different Agile or lean methods, the evolution of these methods is described by Abrahamsson et al (2003). However only a small number is used in industry, with the most common being Scrum, Kanban and Extreme Programming (XP).

Agile terminology for this thesis

The use of Agile terminology within this thesis aligns to that used within published literature which is discussed in Chapter 2.

Agile Software Development (ASD) or Agile development: When the use of the term Agile Software Development is used within this thesis is used as an umbrella term and will be abbreviated to "ASD". References will appropriately be made to specific Agile methods such as Kanban or Scrum.

developer or software developer: Where this term appears in this thesis, it is used as an umbrella term for a developer who's discipline might be titled as a Software Engineer, Web Developer or other. Any other specific roles such as 'front-end developer' will be referenced appropriately.

1.2 Research rationale

This section provides the research rationale behind this thesis and in particular, details of the unique challenges of UX designers and software developers working relationship in the integration of UCD and ASD.

1.2.1 Overview

The rationale behind this thesis came from reviewing relevant literature, conducting preliminary desk research and the researcher becoming aware, through practice-led research, of a disconnect between the UX design and software development roles, their collaboration practices, workflow and techniques for creating online products and services. It became apparent in the preliminary study that without the right integration and collaboration between the ASD process and the UX practices, a detrimental affect occurred to the efficiently and quality, thus the User Experience (UX), of the online product or service.

To ground this observation and provide rationale to this research, initially a broad understanding of how practices and technology have evolved was required via a review of the literature. The literature review (Chapter 2) investigated the UCD and ASD challenges and approaches, including a critical assessment of current reported literature of collaboration practices between UX designers and software developers in an organisational setting. The findings from the literature, which contribute to the rationale of this research, are highlighted below:

1.2.2 Integration of UCD and ASD

Agile methodologies and its variants (Dybå & Dingsøyr, 2008; Inayat, Salim, Marczak, Daneva, & Shamshirband, 2015) are often perceived to be at odds with UCD techniques (Salah et al., 2014). UCD places the user and their requirements at the centre of the design process, aiming to achieve a positive subjective experience and high objective performance with the interface. Researchers and practitioners in UCD have developed methodologies, techniques and processes that enable design teams to create and test solutions before software developers are involved. UCD has been found to positively impact the results of the design and development efforts by reducing customer complaints, training needs, and increasing uptake of resulting products (Bias & Mayhew, 2005).

In a recent systematic literature review (Salah, et al., 2014) on UCD and ASD integration a number of main issues are identified that impact on design and development work. First, ASD promotes the elimination of much of the up-front planning work to remain responsive to changing requirements. This means that there is little time for the usual research, analysis of requirements, persona development or any elaborate prototyping characteristic in UCD. Secondly, another issue is UCD work is not easily divided up into 'chunks' to fit the Agile work practice. The lack of pre-planning of defined design goals makes determining the size of design chunks difficult. Furthermore, designers usually take a more holistic view on the interaction design and information architecture of a website or product, so therefore modularisation and iteratively adding features may be averse to their way of thinking and working.

1.2.3 Collaboration between designers and developers

Salah et al. (2014), also found in their review that the work-dynamic and relationship between designers and developer changes in an ASD setting. This is true in particular for designers, as their job requires them to be "on call" and supply ad-hoc solutions, reviews, and feedback in a team-oriented design process such as ASD. The importance on "working software" as the main yardstick for the design and development progress is a challenge for the designer-developer relationship. So while many studies (Dyba & Dingsøyr, 2008; Salah et al., 2014) address possible barriers to a successful ASD implementation, it is significant there are few studies on how these two crucial roles interact and collaborate.

Brown, Lindgaard and Biddle (2011) observed that much of the interaction time between these roles was used to "re-align" individual work progress to ensure a common understanding of the project aims and ensure product development plans were on track. Ferreira, Sharp and Robinson (2012) found in ethnographic studies that successful integration of Agile and UX work relies on attitudes and work practices such as mutual awareness, expectations about acceptable behaviour, negotiating progress and general engagement with each other. However, as reported in a review by Jurca, Hellman & Maurer (2014b), there is a lack of rigorous insight or evaluation whether and how designers and developers differ in the reported attitudes and practices, and how their co-operation is determined in particular with regards to organisational structures and decision processes.

1.2.4 A changing UX landscape

These days, people are increasingly using multiple computing devices in their daily lives to achieve different tasks and there has been a profound progression and a shift in the facets of computer technology over the past ten years, in particular in the advent of the smartphone.

The proliferation of the devices used and the changes in interfaces means it is now common and often important for users to perform their tasks by interacting with more than one computing platform to achieve interrelated goals (Seffah, Forbrig, & Javahery, 2004). The demand for cross-platform interaction is growing as users adopt multiple devices for different kinds of needs (Santosa & Wigdor, 2013).

The computing industry and the disciplines of HCI and SE have seen the development of the smartphone, wearable devices, ubiquitous and cloud computing, new methods for fabricating 3D materials, and the Internet of Things. Devices are now selected by users based on their form factor to suit their context or needs. For instance, people might tend to read their email or browse the web using their smart phone and use bigger screens (like a desktop or smart TV) to do things like stream on-demand video. Additionally, the devices that people are now using are also changing the way the technology is used and thus designed and developed. An example is with the introduction of the reliable, robust and effective touch screen which became commonplace in 2007 with the Apple iPhone. Since then, a profound shift in user behaviour has been witnessed, away from desktop computers in favour of new form-factor devices with new types of input methods (e.g., touchscreen, voice or augmented reality). The enabling technology has also brought about an entirely new class of Web-enabled applications and architectural ideals.

Fox (2012) compares this shift in culture to the role that public libraries have played in society. During the seventeenth and eighteenth centuries, libraries shifted from being primarily repositories of manuscripts and other intellectual artefacts to having an active role in the dissemination of information. It was not until the nineteenth century that public libraries, a civic presence that we now take for granted, became ubiquitous. This is the context wherein public demands for information increased exponentially that a flourishing of library services took place. Fox compares this proliferation of libraries and services helped to establish an expectation regarding the availability of information in public life. He argues that we are experiencing again a profound shift in the way people access information and library

services with a transition in internet culture where users are no longer bound to their desktops in order to use the Web (Cabage & Zhang, 2013). This has been achieved through the fragmentation of devices that people use to access the internet and accomplish daily tasks that would have normally been done sitting in front of a keyboard and monitor (R. Fox, 2012). This trend will increase towards ubiquitous computing where users will have seamless access to applications regardless of their whereabouts or the computing device at hand (Weiser, 2007)

Consequently, alongside UCD & ASD processes becoming commonplace within industry, especially in large multidisciplinary teams, the methods used to create the online experiences must also learn and adapt to suit the shift in how people use technology. Prior to this shift in the landscape described above, UX for online experiences, was commonly created for the assumption of the desktop computer and mixed SE approaches of Waterfall and some Agile could often exist with UX applied in isolation (Brown, 2013). The shift in the landscape has forced online experiences either in the web or via mobile apps to be reconsidered through techniques like Responsive Web Design (Marcotte, 2010), creating a significant effect on the practices between UX designers and software developers and related studies should now be conducted to further understand the impact this may have had in recent times.

1.2.5 Gap in knowledge

As detailed above, the focus of this thesis is on UX designers and software developers practical working relationship and collaboration practices in integrating UCD and ASD.

The research specifically aims to find out about the determining factors of successful collaboration between designers and developers and how their roles differ in the integration of UCD and the ASD process in an organisational setting.

1.3 Research questions

As described above, the focus of this thesis is on UX designers and software developers practical working relationship and collaboration practices in integrating UCD and ASD.

The overall aim of the research was to gain an understanding of what 'success' looks like between designers and developers working in an Agile environment. This began with the initial overarching question that led into the preliminary study that was the following:

How do UX designers and software developers work together to create online products and services?

This question guided the initial practice-led research and helped to form a focus for further research in combination of a review of the relevant literature, as detailed in Chapter 2. Following this review and the preliminary study, the research evolved to find out about the determining factors of successful collaboration between designers and developers and how their roles differ in the integration of UCD and the ASD process in an organisational setting. In particular, this provided more focused research questions:

How do people integrate UCD with ASD in practice and what is the impact on collaboration between designers and developers in an organisational setting?

This question was answered with a series of in-depth interviews and online survey with designers and developers. The outcomes of the two studies informed the following research question(s):

What does 'success' look like between pairs of designers and developers working in an UCD and ASD environment? What are barriers and what are consequences of successful paired work?

To answer this question, a qualitative approach was selected for two studies. Pairing Study A captured insights from one pair over the course of 1 week and Pairing Study B captured insights using a Contextual Inquiry with 6 design and development pairs who also participated in a Diary Study over the course of 6 weeks to gain further understanding.

1.4 Thesis structure

The outline for the remaining chapters of this thesis is as follows:

Chapter 2: Literature Review – In this chapter a review of the relevant literature for this thesis is presented. This review covers the related empirical work on the integration between UCD and ASD and any previous findings into the relationship and determining factors for successful collaboration between UX designers and software developers in an organisational setting.

Chapter 3: Preliminary Practice-led Research – This chapter presents a preliminary study set within working environment of the researcher, participating as a UX designer. This study gathers an understanding, through a practice-led investigation about the current collaborative practices and perceived problem areas that exist between UX designers and software developers.

Chapter 4: Research Design – This chapter provides an overview of the methodology for the research that was carried out. Including how the key research questions can be answered using a different research approaches and methods to form an overall research design for the empirical studies across a complex organisation.

Chapter 5: Study 1 – Stakeholder Interviews – In this chapter an ethnographically informed study of 9 stakeholders (managers) of UX designers and software developers from different teams across a large organisation is presented. The stakeholders were interviewed for approximately 1 hour each using the in-depth interview approach and the interviews took place over the course of 1 week. This study highlights the organisational and team based challenges for collaboration between designers and developers.

Chapter 6: Study 2 – Online Survey with Designers & Developers – This chapter presents a mixed method study of 109 UX designers and software developers from various teams across a large organisation. The findings highlight that the relationship between UX developers and software designers is an important factor for successful UCD work in an ASD process. However, the determinants and long-term barriers for successful ASD in this relationship are still undetermined and should be explored further alongside new collaboration methods and practices.

Chapter 7: Study 3 – Pairing Study A – In this chapter a pairing study is presented. Insights and experiences of a UX designer and software developer in a large organisation are captured over the period of 1 week via a questionnaire at the beginning and at the end of the week. The findings highlight that success can be found through close collaboration and pairing when the pair produce prototypes together to inform wider team decisions.

Chapter 8: Study 4 – Pairing Study B – This chapter presents an ethnographically informed study of 6 design and development pairings to learn about how they collaborate and work together over the course of 6 weeks. Aiming to understand the determining factors for successful collaboration between the roles in an organisational setting.

Chapter 9: Discussion – In this chapter the findings from the preliminary practice-led study, the literature review and the four research studies are discussed. The studies are discussed in terms of how they answer the main research question(s) and relate to the relevant reported literature.

Chapter 10: Conclusion – In this chapter the thesis is concluded. This includes a final summary of the research, the contributions and how the research could be extended through further work.

Chapter 2
Literature Review

2.1 Introduction

In this chapter a review of the relevant literature for this thesis is presented. This review covers the related empirical work on the integration between UCD and ASD and any previous findings into the relationship and determining factors for successful collaboration between UX designers and software developers in an organisational setting.

The next sections in this chapter describe the literature review objectives (2.2), followed by describing the selected relevant literature (2.3 - 2.8), then the opportunities for further research is discussed (2.9) and finally a summary is then presented (2.10).

2.2 Literature review objectives

The focus of this Literature Review is concerned with the fields of Human-Computer Interaction (HCI) and Software Engineering (SE). In particular, the theories and the previous research relating to the disciplines of design and development within the context of large Agile organisations. It is important to review literature in order to understand the strengths and limitations of different design and development practices and approaches. The aim is also to recognise any controversy, disagreement or change in opinion within the field(s) to discover potential gaps in knowledge, providing encouragement and grounding for further empirical research.

> "The Literature Review is intended to distil information from published sources, capturing the essence of previous research or projects as they might inform the current project." (Hanington & Martin, 2012, p. 112)

As a result of carrying out this literature review, discoveries of potentially new theories and recent research may be made, which will inform the research questions of further studies (Brewer, 2007). The aim is to provide an understanding of the big picture and tie it together with other research and practices that have been previously established. Boyne (2009) establishes the key aims of a literature review and they are depicted below in Figure 2-1.

Brewer (2007) states that a literature review is a useful tool to identify potential gaps of knowledge in the field and to critically analyse existing theory relating to the main research questions.

1. distinguishing what has been done from what needs to be done
2. discovering important variables relevant to the topic
3. synthesizing and gaining a new perspective
4. identifying relationships between ideas and practice
5. establishing the context of the topic or problem
6. rationalizing the significance of the problem
7. enhancing and acquiring the subject vocabulary
8. understanding the structure of the subject
9. relating ideas and theory to applications
10. identifying the main methodologies and research techniques that have been used
11. placing the research in a historical context to show familiarity with state-of-the-art developments

Figure 2-1 - Aims of a Literature Review

2.2.1 Key questions to be answered with this literature review

- What is the relevant background of Human Computer Interaction & Software Engineering?
- What is the background of UCD and ASD and what are the reported integration challenges?
- What are the reported similarities between UCD and ASD?
- What are the reported challenges of integrating UCD and Agile?
- How do UX designers and software developers collaborate? What are the reported challenges and practices?
- What studies have taken place in large organisational settings and where are there opportunities for further work and research?

2.3 Human Computer Interaction & Software Engineering

Human Computer Interaction (HCI) and Software Engineering (SE) have emerged as separate fields and the discussion around how they integrate together has long been discussed and shares important factors with the main focus of this literature review.

The field of HCI researches the design and usage of computer technology, focusing on the interaction between people and computer systems. The term HCI was popularised in 1983 in the book 'The Psychology of Human-Computer Interaction' and the term was first used by the authors in 1980 (Card et al., 1980). Software Engineering (Oettinger, 1967) is the systematic application of engineering to the design, development, operation and maintenance of software. Anthony Oettinger first coined the term and the discipline first emerged to address poor quality software. The aim was to ensure that software was built systematically, measurably, on time, on budget, within specification and with suitable rigour.

Between the two areas, distinctions have been made to show their differences. This is based on the disciplines having different vocabularies (Belenguer, Parra, Torres, & Molina, 2003), different approaches to software development (Ferre & Moreno, 2004) and even a lack of "sound scientific common ground," (Taylor & Coutaz, 1995, p. 1). The differences have naturally developed over time (Grudin & Fielding, 1995) and contributed to a gap between HCI and SE. The considerable literature on combining HCI and SE presents an on-going challenge for researchers and practitioners to find common ground in bringing HCI and SE together. The conclusions of a number of investigations brought to light some of the major integration issues. Law (2003) concludes that the differences are difficult to overcome but can be bridged through a regular dialogue between the two disciplines. Use cases (as described by McMenamin & Palmer, 1984) are explained by Constantine et al. (2003) as the common connection between 'Usage-Centred Design' and SE. Their paper calls for tools that improve on the Unified Modelling Language's (as described by Rumbaugh, Jacobson, & Booch, 1999) lack of constructs for designing a user interface, to support the integration between HCI and SE. A full review of the differences in integration between HCI and SE is beyond the scope of this chapter but a systematic literature review conducted by Bjørnson and Dingsøyr (2008) recognises that closing the gap between HCI and SE lies in the tools, techniques and methods that HCI and SE practitioners and researchers employ.

2.4 User Centred Design and Agile Software Development

At the turn of the millennium two new approaches evolved into HCI and SE – These were User Experience Design (UX) and Agile Software Development (ASD) respectively (E. L. Law & Lárusdóttir, 2015).

Agile Software Development is a philosophy that was put to paper in 2001 (Beck, Beedle, Bennekum, et al., 2001). It emerged because of the need for a lightweight set of software development methods to address drawbacks of heavyweight document-driven software development methodologies—such as Waterfall (Szalvay, 2004). 'Agile' is the summary term for the process where the requirements (e.g., the users' or functional requirements) are addressed by applying solutions iteratively through collaboration between self-organising, cross-functional teams (Dybå & Dingsøyr, 2008).

ASD promotes adaptive planning to counter the perceived shortcomings of traditional plan-driven methodologies, and encourages delivery of early versions of solutions that get continuously improved. This way the team is thought to be able to respond to changing user requirements or business needs. Agile teams strive to deliver an early and fast production of working code and make frequent and incremental changes. This is often achieved through paired programming in short iterative cycles, with contingent user feedback. Importantly, it requires a high degree of collaboration and shared decision-making (Drury-Grogan, L. O'Dwyer, 2013).

Similarly, UX was coined as a term because of limitations presented by usability and human interface design – UX aimed to address a wider scope, covering the design of a person's entire experience and interaction with a system (Alben, 2002; Norman & Nielsen, 2016). It has long been recognised that it is required to consider human capabilities and characteristics when designing technologies and systems. As Nickerson summarised in 1969, when the potential for computer-based technologies was first being fully recognised: "the need for the future is not so much computer oriented people as for people oriented computers" (Nickerson, 1969, p. 178 in the IEEE version).

UX is defined as the perceptions and responses of users that result from their experience of using a product (Garrett, 2011). User Centred Design (UCD) is a set of techniques, methods, procedures and processes as well as a philosophy that places the user at the centre of the

development process (Detweiler, 2007; Gould & Lewis, 1983) The goal of applying UCD is to attempt to satisfy users via producing usable and understandable products that meet their needs and interests. These two terms, 'user experience design' (UX) and 'user centred design' (UCD) are often used interchangeably but there is an important distinction between the two. UX design is a discipline i.e. *what* someone does, whereas UCD can be seen as the process of *how* someone does it. The specifics of both vary and precise definitions can be elusive. UCD is seen as the dominant design process that is applied within UX and by UX designers.

Further background information about UCD and ASD is presented in the Introduction (Chapter 1) of this thesis. The two processes are now commonplace within industry, especially in large multidisciplinary teams. Despite the popularity of the two approaches, Agile methodologies and its variants (Dybå & Dingsøyr, 2008; Inayat et al., 2015) are often perceived to be at odds with UCD techniques (Salah et al., 2014). The integration between the two methods has gained increased interest due to the potential benefits that can be provided by adopted and combining such processes.

Integrating UCD importantly enables developers to understand the needs of users who might use their software and how their work and activities can best support the development of the software to suit the user – the major Agile processes also do not explicitly include guidance for how to develop usable software (Lee, McCrickard, & Stevens, 2009). Secondly, the ASD community often fails in discussion and awareness of users and UX, thus implying either a negligence of UX or focus on less sophisticated UX projects (Aigner, 2009). Furthermore, there is a principled difference between UCD and ASD in terms of the focus, methods and culture that suggests that integration will be fundamentally challenging. A number of researchers have proposed that the combination of UCD and ASD brings about an opportunity to help bridge the gap between HCI and SE and have – Sharp et al. (2004) suggested that this might occur by combining the method of Extreme Programming (XP) and UCD but there is a lack of specific guidance or practical methods.

2.5 Similarities between UCD and ASD

This section focuses on discussing the literature that reports on the similarities and common ground that supports the integration between UCD and ASD.

2.5.1 Focusing on people

UCD and ASD methods both are human centred in their approaches (Lee, 2011). UCD places the user at the centre of the design process and ASD values face-to-face communication and coordination between team members so in theory close work between designers and developers is invaluable in ensuring work remains in sync and on track (Lee et al., 2009). The common focus on people is echoed in the team coherence, which is emphasised by both approaches. ASD uses planning methods to bring the team together (Beck, 1999) and UCD brings people together with the common focus on the user and their goals.

2.5.2 User involvement

Both UCD and ASD methods aim to effectively involve users in the development and iteration of the software. This is achieved in UCD with a variety of different techniques (e.g., usability testing or ethnographical studies). In ASD, one of the most common methods used is Scrum, which has been regarded as a process that emphasises UX because of introducing user involvement through user stories, and by its iterative and communicative nature (Schwaber, 1997). This aspect of Scrum would then concur with the values of many UCD approaches.

Despite this, the UX work is not an obligatory part of the process so user involvement and evaluation is not guaranteed (Salah et al., 2014; Silva da Silva, Selbach Silveira, Maurer, & Hellmann, 2012). How best to involve users represents a challenge in the integration of UCD and ASD that is well reported in the literature but with mixed success especially within the ASD process (Sharp et al., 2004).

2.5.3 Iterative design and development

One of the principles of UCD is that the 'process is iterative' – knowing that the perfect design is unlikely to be achieved at the first attempt and through user involvement and iteration the design can be improved over time. In particular, iterative design attempts to

rectify and learn from problems discovered during usability testing (D. Fox, Sillito, & Maurer, 2008). Likewise, ASD aims to iteratively build working software as a core value to reduce any risk and incorporate regular feedback from "customers" to allow for continuous improvement. In particular, the XP method relies on iterative development and feedback in the form of automated testing and code re-factoring.

Despite the aligned goals of achieving iterative development there are known issues. UCD iteration differs to ASD due to prescribing user involvement at regular intervals of the development of the software (Zahid Hussain, Milchrahm, et al., 2009), whereas in ASD, the focus is automated testing which can be time consuming and difficult to implement (Constantine & Lockwood, 2002b). Likewise, usability testing is completely ignored in ASD methods like XP despite it being part of the aims of the approach (Sharp et al., 2004). The question of how to involve users remains a challenge to practitioners in both the UCD and Agile domain.

2.6 UCD & ASD integration challenges

This section focuses on discussing the literature that reports on the challenges of integrating the UCD and ASD approaches.

Without aiming to provide an exhaustive list, Table 2-1 summarises some of the main challenges of ASD and UCD integration (and thus implications for designers and developers) as identified by some major recent review articles in this area (Caballero et al., 2016; Brhel et al., 2015; Da Silva et al., 2012; Law & Larusdottir, 2015; Salah et al., 2014).

ASD	**UCD**	**Solutions suggested**
Lack of allocated time: Delivering working code quickly, focus on functionality	Upfront planning activities: Requires insight, research and design	Separate predevelopment phase (sprint 0) called "upfront design";
	(see Salah et al, 2014; Cabalerro et al., 2016)	
Work divided into chunks: Tight deadlines	Designing for the whole user experience; holistic design of UCD	Flexible chunking (or time-boxing) of design activities; well-defined design goals
	(see Salah et al., 2014 and Law & Larusdottir, 2015)	
Working software over comprehensive documentation Fast-paced releases; deadlines	Medium to long-term studies before implementation work; prototypes; Usability testing;	"Upfront design"; discount usability evaluation; UX designers as surrogate users
	(see Salah et al., 2014)	
Lack of documentation	Decisions based on information; data, reports, prototypes	Artifact-based; web/wiki-based documentation; frequent interaction
	(see Salah et al., 2014 and Kollmann, 2008)	
Limitation of work-in-progress	Delays UX designer from giving effective feedback on design	None
	(see Law & Larusdottir, 2015)	
Decisions are made quickly project manager is not the accountable decision maker	Decisions after data gathering and thorough analysis, iterative design	Communicating design vision early and frequently; ad hoc meetings
	(see Drury & O'Dwyer, 2013)	
Cross-functional teams	UX designer often in a specialist centralized team/services	UX designer to be co-located with developers and team
	(see Brhel, 2015)	

Using tools/default metrics to measure work progress	UX professionals cannot easily track the interplay between user evaluation and redesign (see Law & Larusdottir, 2015)	None

Table 2—1: UCD and ASD integration challenges & conflicts and suggested solutions

2.6.1 Lack of time for up-front planning and design activities

One of the main reported issues in the integration of UCD and ASD is that ASD promotes the elimination of much of the up-front planning work to remain responsive to changing requirements (Ferreira, Noble, & Biddle, 2007c; Lee et al., 2009). This means that there is little time for the usual research, analysis of requirements or any elaborate prototyping work that is typically adopted in UCD.

Using Agile methods, the focus is on delivering working releases little and often (Lievesley & Yee, 2006; Singh, 2008) over spending time on design planning activities (Federoff & Courage, 2009; J Kollmann, 2008). This further compounds the lack of time available for UCD methods (Chamberlain, Sharp, & Maiden, 2006) such as design research to discover and evaluate new problems to then ideate around new speculative solutions (Lievesley & Yee, 2006; Ungar & White, 2008).

This is often addressed by doing upfront design separately to the development process. In particular, researchers and practitioners in UCD have developed methodologies, techniques and processes that enable design teams to create, prototype and test solutions before developers are involved. UCD has been found to positively impact the results of the design and development efforts by reducing customer complaints, training needs, and increasing uptake of resulting products (Bias & Mayhew, 2005).

Conducting UX design ahead of development work like this helps to plan out and provide a comprehensive view of the system or service being created (Chamberlain et al., 2006). This upfront design stage is also often also referred to as Iteration 0 or Sprint 0.

One argument is that upfront design like this provides benefits in reducing poor design decisions that might occur without thorough investigation, helping to reduce problems later on (Ferreira et al., 2007c). This type of up-front work is also found to be used by development and quality assurance disciplines to work on complex features which might have a high development cost and initial upfront work is required to make decisions about the way forward (Miller, 2005). Constantine (2002a, p. 8) 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, p. 6) also agree that "Emergent Design doesn't work very well for user interfaces." and propose

that “Some Design Up Front seems to provide better guidance to the development team and provides earlier opportunities for feedback.”

Despite this, accounts in the literature about the amount of up-front design that should take place are often conflicting. Sy and Miller (2008) advise that UX designers should work one iteration ahead of the software developers. However, Ungar and White (2008), recommend completing enough up-front design for two or three Agile iterations worth of design implementation. Additionally, in the preliminary practice-led study, the amount of up-front design caused frustrations due to the heavy handovers that subsequently occurred. What is clear in the UCD & ASD literature however, is that with up-front design, and how much is enough, remains controversial (Adikari, McDonald, & Campbell, 2009).

2.6.2 UCD work is not easily divided up in ‘chunks’

The UCD principle of ‘designing for the whole user experience’ is not compatible with the incremental nature of Agile from the outset. In ASD work is divided up into chunks and smaller releases in a “feature by feature” way, often known as “slices” so that releases can occur frequently with low risk. This breaking down of work in to smaller parts makes it difficult to consider the holistic design at all times, contradicting with UCD practices (Sy, 2007).

Design ‘chunking’ helps to break down design into cycle sized parts that incrementally add features to the overall interface design (Miller, 2005). When this is interlaced with the incremental nature of the ASD it is reported to be more difficult for it to work and often designers can find this difficult to grasp as it is adverse to their normal procedures and way of thinking (Sy, 2007).

Salah et al (2014) reports this to be due to three main reasons: Firstly, it’s difficult to appropriately predict the size of design chunk in comparison to the development iteration (Sy & Miller, 2008). Secondly, it’s hard to maintain how different design chunks are dependent or independent of each other (Sy, 2007). Thirdly, there is difficulty in understanding the UCD activities that are broader in scope and may require more work across the entire system or interface (Hodgetts, 2005). The problems found in design chunking have been attempted to be addressed with a number of different practices, including: Making sure there are well defined design goals in the project, mapping the chunks to features of the system (Najafi &

Toyoshiba, 2008) and as Hodgetts (2005) discusses, by reducing the time consuming UCD activities that could occur at a different stage e.g. Sprint 0.

Overall, it is reported that this challenge frequently presents results in the UX that is disjointed, piecemeal and lacks an overall structure and vision that the UX designer desires (Albisetti, 2010; Lee, 2011; Obendorf & Finck, 2008).

2.6.3 Conducting usability testing and gathering feedback

In UCD, as part of the principle in gaining an 'explicit understanding of users, tasks and environments' – it is required to work with the users or potential users of a system over time to make sense of their varying emotions, abilities, circumstances in their lives. To do this often ethnographical, medium to long term studies are prescribed alongside typical users performance measurements via usability or guerrilla testing methods (Z Hussain et al., 2012; Preece, Rogers, & Sharp, 2002). Due to the fast-paced release driven nature that Agile and Lean approaches uphold, the literature reports of a number of challenges that this presents, these include: The research methods for testing, the scheduling of the testing, recruitment of the users and the high-cost of running the sessions.

To evaluate and test prototypes or working builds, methods of research or usability testing often require time consuming scheduling with representative end users of the product or system. This often poses a challenge (Federoff & Courage, 2009; Lee & McCrickard, 2007) and conflicts with the time boxed nature of Agile so as a result often is reduced in scope, done without representative users or missed completely. This undermines the UCD process and thus risks the quality and usability of the experience. The scheduling is often unpredictable and presents a lack of clarity in regards to the timing of the evaluations as part of the iterative, continuous development of ASD (Ferreira, Sharp, & Robinson, 2012b).

In the literature the challenge of tight timeframes and scheduling for user research is reportedly overcome via various methods: Firstly by finding more preparation time in advance for user research (Illmensee & Muff, 2009), utilising techniques such as heuristic evaluation (D. Fox et al., 2008; Memmel, Gundelsweiler, & Reiterer, 2007) and the Rapid Iterative Testing and Evaluation (RITE) method (Dayton & Barnum, 2009; Federoff & Courage, 2009) and also Fox et al (2008) reports using lower fidelity prototypes to conduct usability tests. Some researchers also suggest fitting usability testing into the context of other

ASD tests, for example, acceptance-testing sessions (Ferreira, Noble, & Biddle, 2007a), as it can fit in with the completion of iterations and releases. Armitage (2004) and Wilcox (2007) report that feedback from regular releases of the product can serve the same purpose as organised usability testing.

As noted above, the compressed ASD timescales also pose a challenge in accessing and recruiting representative users for design research (Detweiler, 2007; Ferreira et al., 2007a). This is often down to planning requirements that can often take weeks depending on the factors involved in the research. This might not fit with the ASD schedules and is reportedly often overcome with planning user inclusion in advance (Kollmann, 2008), accessing an existing panel of users via recruitment companies who can frequently schedule testing (Illmensee & Muff, 2009), doing collaborative peer reviews (Memmel, Gundelsweiler, et al., 2007) or lastly, Detweiler reports using remote panel testing to reduce the challenges presented by localized recruitment needs.

Despite these reported practices of including usability testing methods into the ASD process, there are still challenges that are not addressed in the literature to a) incorporate more holistic ethnographical design research into the process and to b) incorporate the results of testing and research regularly into the development cycles. Agile teams reported a lack of time to respond to results of usability evaluations and feedback (Detweiler, 2007; Miller & Sy, 2009). However, this was reported to be addressed by introducing dedicated cycles for incorporating user feedback (McInerney & Maurer, 2005) and also Millar & Sy reported using a UX designer to help plan and ensure user feedback was being incorporated by validating designs that were handed over to developers to implement.

2.6.4 UCD practitioner representation in Agile teams

In Agile teams is it common to have a mixture of disciplines but the representation of UX designers often varies in comparison to other roles. This results in a UCD staff shortage (e.g., UX specialists from a centralised UX department have to work for several teams simultaneously). This can negatively impact the quality and product usability or UX. Brown et al (2011) declared that the position of the design role in Agile teams is ill defined. Ferreira et al. (2007b) stated that interaction designers being present in XP teams is essential for achieving good UX, whereas having developers doing design work is not ideal. Despite this, although Ferreira et al. (2007b) reported that the UCD role should be represented in Agile

teams, it is most important for the UCD specialist to adopt the Agile mind-set and adapt to an iterative, fast and feedback based development process rather than adopting big upfront design.

Despite the literature supporting the consistent representation of a UCD practitioner in an Agile team, this representation does vary between different organisations. In organisations where the UCD practitioners are shared among a number of teams it can lead to extra burden upon the role. Federoff and Courage (2009) reported the results of a survey that was conducted after 30 teams transitioned to Agile. Only 30% of the 'development services' team that included members from documentation, usability and UX design believed that their teams are more effective after transitioning to Agile. The reasons for this was because they were assigned to too many teams, attended lengthy meetings, and the tighter Agile schedules did not provide enough time to finish their work. Furthermore, the frequent context switching resulted in a severe productivity loss for the team members. So mainly the reasons were attributed to lack of resources since having the development services team members shared among a number of teams was no longer effective (Federoff & Courage, 2009).

Hussain et al. (2009) reported the results of an online survey with 92 respondents who were UCD professionals and software developers. 52% of respondents preferred having a dedicated UCD professional, 4% indicated that a developer with interest in HCI/usability performs this role and 36% gave no answer. One respondent indicated that "you cannot get good UI with a developer doing it, they will always have a conflict of interest focusing on what is easier to build rather than what is the best user experience". Another respondent indicated that "having a dedicated UCD person is always better, very few developers I have seen with an interest in UCD are actually able to focus on the user rather than just 'cool design' or 'what is easier to develop'". Some respondents supported either scenario: "Either can work: it depends on the skill of the person, knowledge of the domain and interpersonal team dynamics" (Zahid Hussain, Slany, et al., 2009). In supporting this, Fox et al. (2008) reported that a lack of UCD specialist can allow a developer to act as UCD specialist and as a result can deal with any usability related issue instead of losing time by passing it to another team.

The literature reports of a number of practices that can be used when dealing with the increased work upon UCD practitioners, for example, conducting mentoring process to

developers so as they can perform the role of UCD practitioner (Albisetti, 2010), office hours (Federoff & Courage, 2009), decreasing UCD practitioner workload (Leszek & Courage, 2008) and distributing UCD practitioner workload on a UCD researcher and UCD prototype specialist (Williams & Ferguson, 2007). Moreover, in the online survey conducted by Hussain et al. (2009) they found that 75% of respondents believed developers can pick up HCI skills by pairing with a UCD professional, 66% mentioned that this can be achieved via training. Albisetti (2010) reported choosing a number of developers who had an interest in UCD and mentoring them in performing UI reviews for other developers. Albisetti found that this resulted in better productivity, less workload for the UCD practitioner, more experience gained by developers with UI reviews that resulted in improved work quality (Albisetti, 2010).

2.7 Differences between designers and developers

The UCD & ASD literature describes UX designers and software developers as distinct, even clashing groups of practitioners. As described by Memmel et al (2007) – "HCI and SE are recognised as professions made up of very distinct populations." In which these groups have their own concerns, aims and approaches for designing and engineering software and becomes part of the challenge of combining UCD with ASD. UCD practitioners are depicted as unknowingly thrust into interactions with a strange band of people and then 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 & Khan, 2010, pp. 32–38).

Maudet (2017) described UX designers and software developers of interactive software products as having very different backgrounds and skills, where they focus on different aspects of the design and development process (Löwgren, 1995). Designers often communicate visually: They use graphical editors, e.g. Adobe Illustrator and Photoshop, to create "static design documents" (Newman & Landay, 2000) such as UX wireframes and design mock-ups. They will prioritise visual appearance and interaction behaviour (Cooper, Reimann, & Cronin, 2007) over the rules and data structures that govern software development. By contrast, developers work with abstractions: They use text editors and Integrated Development Environments (IDEs) to create functional systems. In doing so developers prioritise the translation of design documents into implementable formats over the

details of visual design and user interaction (Maudet et al., 2017). Despite this, the two groups are found in the literature to not be as distinct in practice as some report. Studies of practice have found that those who carry out the UX design work may be software developers with an interest or training in UX design (Ferreira et al., 2007a).

Based on a review of the UCD & ASD literature, da Silva et al. (2011) insist that the outlook is positive and despite some of the cited differences between the groups, improvements can be found if they can find ways to work together - "Agile Methods have a distinct culture that at first glance seems to conflict with User-Centred Design (UCD) (McInerney & Maurer, 2005). However, according to these same authors, the use of Agile methods can result in improved usability." Wolkerstorfer et al. (2008) and Patton (2005) describe how combining these different skills, by way of combining UX design and software development, achieved various aims for their ASD 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-centred design process (usable, accessible and accepted products, end-user integration)" (Wolkerstorfer et al., 2008). Patton (2005) describes how both approaches complemented each other to improve software quality and user satisfaction - "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, 2005).

This section shows that the UCD & ASD literature presents the differences between UX designers and software developers as competing and clashing. Yet, working together has advantages for teams who can overcome the differences. There are few reports on how the differences can be overcome in different settings and how specific interactions and collaborative practices between UX designers and software developers can help to integrate the two roles.

2.8 Collaboration challenges between designers and developers

Salah et al. (2014) found in their review that the work-dynamic and relationship between designers and developers to have changed in an ASD setting (Ferreira et al., 2007a). This is true in particular for designers, as their job requires them to be "on call" and supply ad-hoc

solutions (McInerney & Maurer, 2005), reviews, and feedback in a team-oriented design process such as ASD. This is demonstrated in Agile as is it common to have tight deadlines (e.g. two week sprints) so the ability to keep users involved in the evaluation of the product or experience as it is designed iteratively (a key aspect of UCD) becomes more challenging and can end up being cut short. Furthermore, the importance on "working software" as the main yardstick for the design and development progress is a challenge for the designer-developer relationship. The highly compressed time scales and reliance on team self governance of Agile development processes also require more active involvement from UX managers or stakeholders to ensure the regular UCD activities in ASD planning and scheduling (Detweiler, 2007). Additionally, the Agile principle which states that "Simplicity–the art of maximizing the amount of work not done–is essential" represents a challenge to integrating UCD and ASD since simplicity in the user interface does not always mean that the implementation is also simple and vice versa (Lee et al., 2009). On-going and continuous communication needs to be maintained between UCD practitioners and software developers to avoid the occurrence of delays and bottle necks in the development process (Ferreira et al., 2007b).

A number collaboration challenges and practices that affect the work dynamics between designers and developers are found in the literature. These challenges and practices will be discussed in the following subsections

2.8.1 Sharing an understanding of users

Alongside the standards definition (ISO 9241-210: 2010) for UCD and the principle of *understanding of users, tasks and environments* it is reported that empathising with the end user and their needs is a necessary understanding for both designers and developers to create a quality UX (Kollmann, 2008). Its important to invest time in making sure the entire team understands and agrees on the target audience results, objectives, Key Performance Indicators (KPIs) or Objectives and Key Results (OKRs) to ease the collection of insights and on-going required user feedback iterations into the development process. This is also reported to help UX designers to stay true to their holistic vision and enable them to make decisions on feature sets and design trajectories (Miller, 2005).

2.8.2 Sharing an understanding of design vision

Ambler (2008) reports that collaboration between UX designers and developers should be supported by widely communicating the design "vision" – including the intentions, goals and the rationale. Without this communication to the development team, the vision is useless (J Kollmann, 2008). The sharing of the design vision minimises rework and illuminates any integration issues early on in the process (Ungar & White, 2008), allowing team member to familiarise themselves with the key design goals of the UX in order to ease any future challenges or decision-making. Furthermore, setting holistic design goals amongst the team allows everyone to have a shared understanding of important aspects of the UX from the users perspective and this will have a knock on effect if the developers later need to prioritise work e.g. bug fixes (Lee et al., 2009). Literature also reports that early externalisation of the design vision to stakeholders should be encouraged as part of improving the potential success of the project (Memmel, Reiterer, & Holzinger, 2007).

This sharing and understanding of the design vision was reported to be achieved via a number of methods that included: the design studio (Ungar & White, 2008), engaging developers in multiple design options (McInerney & Maurer, 2005), developers taking part in UI specifications (Albisetti, 2010), sharing design artefacts and prototypes (Brown et al., 2011; Lee & McCrickard, 2007), and utilising information radiators (Kollmann, 2008). Despite these recommendations the separation of the development team and wider stakeholders has not necessarily been addressed. With UX practitioners commonly spending time working independently from developers on the design vision in order to then communicate it to get buy-in and awareness (Memmel, Reiterer, et al., 2007).

2.8.3 UX designers and software developers staying in sync

UX practitioners reported in the literature that the lack of communication of frequent changes in the ASD process caused a lot of confusion and required an immense effort from the UX team to handle the changes in addition to struggling to remain on track with the development team schedule (Budwig, Jeong, & Kelkar, 2009).

When the two disciplines are not in sync with each other due to a lack of collaboration or coordination (Detweiler, 2007) it can mean that the UX design and development efforts are combined to create what is described in the literature as a "design drift" in the implemented

system. In this case, the design and subsequently the usability may be undermined due to the lack of coordination of the different roles, causing frustration and poor ultimately poor UX.

To reduce the chance of this occurring, synchronisation points are needed to allow for close collaboration that will keep information flowing and a dialogue between all disciplines involved in the project. This is supposedly addressed via the attendance of the UX team in daily scrums and stand ups (Budwig, 2009; Najafi & Toyoshiba, 2008) to increase the visibility of UX team's work (Lee, Judge, & McCrickard, 2011) and daily communication of UX designers to clarify design and inform the developers about additions or changes required for the UI (Albisetti, 2010).

2.8.4 Refining the level of documentation

ASD methods aim to achieve minimal documentation in favour of reducing time and releasing working software. Additionally, more frequent interaction with people reduces documentation and thus Agile team members perceive documentation as insufficient for communicating interactive behaviour (Kollmann, 2008). Despite this, the literature argues that to properly integrate Agile and UCD, documentation is crucial for estimation and implementation. Furthermore, the lack of proper requirements documentation including the "design vision" was reported to lead to confusion in regards to UX deliverables (Budwig et al., 2009).

It is argued that a variety of integration related issues can and should be reduced with documentation, including; Firstly, documenting design rationale in order to justify and record prior design decisions (McInerney & Maurer, 2005). Secondly, recording and documenting user requirements and relevant user insights as it can affect the decision-making when creating or changing user stories (Moreno & Yagüe, 2012). Thirdly, there is a need to document the latest designs, expected delivery date and design research and testing results and recommendation for fixes (Sy, 2007). Salah et al. found in their review that the challenge of a lack of documentation was addressed through documenting via several techniques. Obendorf & Finck (2008) suggested documenting through wikis and user scenarios. Kollmann (2008) suggests documenting via personas, sketches, wireframes, information radiators and prototypes. Williams & Ferguson (2007) via webpages, via use cases (Detweiler, 2007), design patterns (McInerney & Maurer, 2005) and via tool support (Moreno & Yagüe, 2012).

2.8.5 Working with the right tools

One of the most crucial problems to the integration of UCD and ASD, as reported by Coatta and Rutter (2011), is the lack of sufficient development tools. A lack of efficient tooling leads to the inability to modify user interfaces across an interactive system without impacting the underlying code. Furthermore, the difficulty in tooling makes developers more reluctant to agree on modifications to the UX that might be suggested by the UCD practitioners, making the task of convincing and justifying changes to the user interface a challenge. Additionally, Coatta and Rutter (2011) stated that the different tools used by UCD practitioners and software developers can create or add to communication problems.

To improve this situation, there are examples in the literature of how tooling is used to help bridge the reported gap between designers and developers. Several authors propose extending engineering representations to also encompass interaction design. For example, Unified Modelling Language (UML[2]) diagrams can be extended to describe and cater for interactive behaviours (Pinheiro, Silva, & Paton, 2000) and serve as a communication tool between designers and developers. Also, a common approach and inspired by the popularity of software design patterns (Gamma, Helm, Johnson, & Vlissides, 2002), Borchers (Borchers, 2001) proposed the creation of catalogues of interaction design patterns, and then later Wiemann (2016) proposed using them as the 'Lingua Franca' (Erickson, 2000) between UX designers and software developers. Often today, designers and developers use the well-established design patterns defined by popular software vendors, such as Apple's iOS Human-Interface Guidelines[3] and Google's Material Design[4]. Although these provide a useful starting point for understanding and implementing the conventions of a particular platform, they cannot solve all representation issues, especially when creating new interactions.

Another reported approach, as described by Maudet et al (2017) is to bridge the activity of prototyping with the software development activity of coding. From the as early as the mid

[2] http://www.uml.org/

[3] https://developer.apple.com/design/

[4] https://material.io/design/

1990s, Myers (1995) advocated prototyping tools that support a wide range of interactions. Landay et al. (1995) proposed an electronic sketching approach and envisioned "a future in which the user interface code will be generated by user interface designers using tools" (Landay & Myers, 1995, pp. 43–50). Within the past few years, to help bridge the reported gap in tooling between design artefacts and the subsequent implementation, numerous commercial prototyping tools have emerged, and in a 2015 survey of 4,000 designers found that 53% use a prototyping tool (Vinh, 2015). However, these tools do not result in final implementations: it is reported in the literature that designers still lack tools that link their representations to developer's implementations (Maudet et al., 2017). Moreno and Yagüe (2012) declare that improved tooling could be the solution to the lack of usability expertise of customers, users and software developers.

2.8.6 Timing & scheduling activities with each other

How UX designers and Agile developers can coordinate their activities with each other is an important factor for integrating UCD and ASD, especially when it comes to the timing and scheduling of their activities. When software developers implement and write code, all activities from architecture through design, programming and testing within a single iteration usually need to take place (Hodgetts, 2005), and questions that affect the UX design may be easily resolved if the UX designer(s) are on hand during the development process (Ferreira et al., 2007a; Wolkerstorfer et al., 2008). As reported by Ferreira (2007a), a problem with this strategy is that developers do not always approach UX designers at the required moment when input is required and the issue might then be forgotten or deprioritised. Coordinating development work with other 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. In addition, the literature suggests that a popular view in practice is to adapt UX design techniques to better fit with an Agile development approach (Detweiler, 2007; Federoff & Courage, 2009). There are reports on how UX design techniques can be adapted to be more lightweight, in order to fit with the short Agile iterations (Beyer, Holtzblatt, & Baker, 2004; Constantine & Lockwood, 2002a; Sy, 2007).

There are also reports that provide 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 who are working in an Agile environment or team - 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 Vee (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-centred designer working within Agile needs to adapt quickly as perceived business value changes."

2.8.7 Co-location of the UX designers and software developers

In the literature review by Salah et al (2014), location of designers and developers and their respective teams is reported as a key factor in the integration of UCD and ASD. McInerney and Maurer (2005) describes the possible impact of the ASD process on UCD in that UCD becomes more of a team effort and the UX designer needs to be "on call" to participate in discussions that are ad-hoc in nature. Furthermore, Williams and Ferguson (2007) report that co-location is perceived as best practice by many practitioners since it simplifies collaboration together and permits continuous communication, negotiation, knowledge sharing and enables instant decision-making between developers and designers (Tzanidou & Ferreira, 2010). Hussain et al. (2009) also stated that colocation makes it easier to influence the design as it progresses and helps the UX designer to become integrated within the team since they are available to answer questions and address problems that arise during the iteration. Fox et al. (2008) reported that in case of co-location of UCD practitioners and developers the exchange of design is constant and on going. Co-location allows developers to be aware of designers work and uncover design areas that could cause problems in the implementation (Tzanidou & Ferreira, 2010).

Ferreira et al. (2010) also stated that teams who support co-location between UCD practitioners and software developers promote favourable environments where stakeholders interact continuously to achieve work progress. Lee et al. (2011) identified that a challenge for integrating Agile and UCD in a distributed environment is to effectively support the sharing of documents and artefacts among team members who are physically separated.

Fox et al. (2008) reported that in the case of non co-location of UCD practitioners and developers the exchange of design got delayed. Additionally, Sy and Miller (2008) described that non co-location of the team has a number of consequences, for example, introducing time and language barriers, difficulty of communication (Albisetti, 2010), lack of sense of

team, and lack of trust (Sy & Miller, 2008). Furthermore, Budwig et al. (2009) reported difficulties faced by the UX designers in coordinating with external (e.g., off-shore) development teams and as a result development and UX teams are co-located to decrease efforts of coordinating the work of remote teams (Budwig et al., 2009). Also, Najafi and Toyoshiba (2008) reported that the geographical separation of the UX team from the engineering team led to the exclusion of the UX teams from release planning, sprints and Scrum meetings. Also it led to a lack of knowledge of the implemented features per sprint, an inability to conduct user research or testing of design detailed specifications and difficulty in understanding opportunities and constraints in terms of design options and understanding the bigger picture (Najafi & Toyoshiba, 2008).

Despite of all the acknowledged benefits of co-location, Lievesley and Yee (2006) refused to co-locate designers with the development team. This was due to a number of reported issues - the need for designers at the initial iterations to employ extensive mental efforts to make sense of and synthesise diverse user interests, information and influences. They report that would have not been possible to accomplish in an unfamiliar and tension laden environment with tactical and operational noise that could impede the designers' ability to envision the completed software product and position it in relation to the end user's requirements. In this case the design team worked independently of the development team in early creative stages, but rigorous communication methods were employed to deal with the physical separation (Lievesley & Yee, 2006).

The research shows that location is a key factor in the integration of UCD and ASD, with benefits often being found through the co-location of UX designers and software developers. When co-location is not the case, Ferreira et al. (2011) suggests it can be credited to distinct work group cultures and organisational policies that shaped their cooperation views upon the roles. Ferreira et al. (2012a) stated that integrating UX design and Agile is shaped via a complex interplay of organisational and team commitments that determines how responsibilities are divided between developers and designers, how different values are placed on different roles' contributions and the working rhythm that has been established between each group. Despite this challenge, understanding the methods that designers and developers employ in different organisational settings and circumstances is lacking in recent research studies.

2.9 Opportunities for further work

In the literature, studies of UCD and ASD are heavily influenced by practice but few studies have investigated the settings and contributing factors of practitioners work. As a result, how UX designers and developers work together in different settings is largely unexplored and there is a scarcity of studies across organisational settings.

The few existing studies that have taken account of organisational settings have observed practitioners working in everyday contexts. The framework for integrating Agile development and user-centred design proposed by Chamberlain et al. (2006) addressed challenges from practice based on observations of UX designers and developers. Kollmann et al. (2009) observed UX design practitioners working with Agile teams and determined the importance of identity and vision in successful UX design. Ferreira et al. (2010) identified how the role of values and assumptions regarding the best way to create software, shapes the integration of UCD with ASD. Brown et al. (2008) shadowed practitioners to ascertain the types of artefacts that developers and designers use to communicate. Lastly, Ferreira (2012b) reports on three ethnographically-informed field studies of the day-to-day practice of developers and designers in an organisational setting. The results show that integration is achieved in practice through (1) mutual awareness, (2) expectations about acceptable behaviour, (3) negotiating progress and (4) engaging with each other. Successful integration relies on practices that support and maintain these four aspects in the day-to-day work of developers and designers.

While these studies take account of the circumstances surrounding Agile development and UX design integration in practice, the focus of these studies vary widely and are still rare in recent times. Furthermore, the integration of UCD and ASD is an on going area of investigation and research. Some of the key factors for challenges in this integration are reported to be due to conflicting methods that both approaches uphold. However, many studies have found ways for the gap in approaches to be bridged with different methods. These include methods to keep the users involvement throughout the ASD process, iterating whilst keeping in mind the holistic vision and planning in time "upfront" for creating a "design vision".

Despite this, in the preliminary practice-led research, the outcomes and success factors varied wildly based on some key differences in the setting and structural differences of the projects.

The factors that contribute to these sorts of differences and the effect on collaborative practices between designers and developers in large organisations remain largely unexplored.

2.10 Summary

The need for better integration between Human Computer Interaction (HCI) and Software Engineering (SE) has existed for many years and in reviewing literature and recent systematic literature reviews (SLRs) shows that integrating UCD and Agile extends this discussion.

As discussed, in the literature there has been on-going research and systematic literature reviews that have looked at how UX and Agile combine (Budwig, 2009) but there is very little research in recent times about the specific collaboration methodologies between people working in a design role and the people working in an engineering role. Furthermore, there appears to be a clear gap in knowledge of what successful collaboration looks like between UX designers and software developers and specifically what practices contribute towards a truly successful online experience.

To date many studies (Dyba & Dingsøyr, 2008; Salah et al., 2014) address possible barriers to a successful ASD implementation, but there are only few studies on how these two crucial roles interact and collaborate. Brown, Lindgaard and Biddle (2011) observed that much of the interaction time between these roles was used to "re-align" individual work progress to ensure a common understanding of the project aims and ensure product development plans were on track.

Ferreira (2012b) found in ethnographic studies that successful integration of Agile and UX work relies on attitudes and work practices such as mutual awareness, expectations about acceptable behaviour, negotiating progress and general engagement with each other. But there is a lack of rigorous insight or evaluation (see a review by Jurca, Hellman, & Maurer, 2014) whether and how designers and developers differ in their attitudes and practices, and how their co-operation is determined by organisational structures and decision-making processes.

With this in mind, the next chapters aim to enhance the findings of this Literature Review by conducting several empirical studies to investigate the current industrial practices for integrating UCD and ASD in a complex organisational setting.

Chapter 3
Preliminary Practice-led Research

3.1 Introduction

This chapter presents an ethnographically informed case study of a product team working in a large UK based media organisation. The observations in this study covered a period of 4 months while the researcher was working as a UX designer within the product team. The observations covered the course of two projects across the product team. The purpose of this study as a preliminary practice-led study was to inform the researcher of areas for further research and improvements across the two fields of HCI and SE alongside analysing the current empirical work as discussed in the Literature Review.

The next section (3.2) presents the background information about how this study came about, the people involved and the projects being worked on. After that the data collection and analysis methods are detailed (3.3). Next, the themes that emerged from the study are discussed (3.4) and then these findings are discussed (3.5) and finally the key findings from the work is summarised (3.6).

3.2 Background

This study emerged from the practical circumstances and natural context of the researcher's working environment; a large organisation with several large product development teams that have all adopted ASD. For the team in this study, ASD processes are being followed using the Scrum method alongside a UCD approach to create an online user experience for large and varying audiences around the UK.

The role of the researcher working within the team was primarily to work and perform as a UX designer but alongside this there was the opportunity observe and capture salient insights into the practices and processes that the team and disciplines adopt to inform further research.

In particular, the aim was to observe factors that were either considered part of the UX design role or somehow considerably affected the UX design role of the researcher including how the a designer works together with software developers. To summarise this, the following describes the main research question for this study: *How do UX designers and software developers work together to create online products and services?*

The Literature Review (Chapter 2) of the relevant fields was on going during the time of this study and following the initial review of the relevant reports, it was noted that current research had placed UCD at odds with ASD methodologies and integration between the two areas was well debated. However, there was an apparent lack of research on how the two crucial roles of a UX designer and software developer interact and collaborate. Similarly, UX practitioners in the researcher's organisation had anecdotally talked about difficulties in striking the right balance on projects when working with developers in ASD teams.

These initial insights and literature research guided this preliminary study and the projects involved would ideally provide enough of a representative example of common practices and processes to highlight areas of potential further research and investigation.

3.2.1 People

As described above, the role of the researcher in this study was to work as a UX Designer on an online product in the organisation, and during the course of this study the researcher worked on two different web-based projects. This role of a practice-led researcher as a participant observer provides a unique perspective and insight into the methods and practices that are operating and evolving in an environment of this type. As a UX designer, the researcher was embedded within the UX team that was organised separately from the ASD team. The role of a practice-led researcher in this study is discussed more thoroughly in the Research Design (Chapter 4). It is important to acknowledge that despite the limitations presented by possible bias the situation presented a good opportunity to begin the research in a naturalistic setting.

Alongside the researcher, the colleagues in the surrounding team were the participants during this study. The roles within the wider team included other UX Designers, Developers (Software Engineers & Web Developers), Business Analysts, Project Managers, Technical Leads & Architects, Testers and Product Owners. There were 6 individuals working within

the ASD team who were classed as developers and all had varying degrees of experience of Agile methods and their role.

3.2.2 Projects

At the time of the observations, there was one major project that was on going in the researcher's team. This was to create a new website that would over time start to replace the current online proposition for one of the organisation's online products. The rationale to do this piece of work came from the observed change in audience behaviours and the proliferation in mobile device usage to access the organisations web-based services. It was decided that a website which catered solely for a desktop experience would no longer be suitable and instead a single responsive website would replace it. This was a practice that was being adopted by many similar large organisations around the world through the techniques of Responsive Web Design (Marcotte, 2010). During the observations, this work manifested itself into two main projects, which are described below:

The first project, *Project 1,* involved the development of different features of the new website and the observations took place over the course of two months. In this particular project the UX designers were not dedicated to a single part of the project, but instead worked together as a team and were spread across various aspects of the work.

Project 2 evolved out of the changing requirements of some of the features in *Project 1*. In this project the researcher had the opportunity to work directly alongside a developer from the team. Both the researcher and the developer were solely concentrated on this sub-project due to looming deadlines on the work and the reduced amount of other UX work available at the time. They were both embedded within the same ASD team and they worked together on the feature from the beginning until it was released on the product.

3.3 Data Collection & Analysis

Qualitative methods were employed to gain an insight and understand about the participants and their work. Relevant observations, events, interactions, reactions, opinions and problems were noted down by the participant observer and were recorded at regular intervals throughout the projects. The observations across both projects took place over the course of 4 months. This spanned the entire duration of the researchers involvement as a UX designer in

the product. The aim was to capture salient episodes as they happened and this was done via periodical notes written down in a notebook. In particular, notes were made about the following:

- How the UX design role would come together with developers and how collaboration would occur.
- Descriptions of context, location and movements, for example whether people moved between desks to collaborate with the other role.
- How processes surrounding the UX designer would effect how decision-making occurred with the wider team and stakeholders (line-managers).
- Outcomes of the work including how the UX design and development roles contributed to different parts of the project or the experience.

A thematic content analysis of the observations was applied to establish categories and themes that reflected the outcomes of the research this would then inform the further study by the researcher.

3.4 Findings

As discussed, the aim of this preliminary practice-led research was to inform and focus in on an area for further research alongside the insight sought from the review of the relevant review of the literature (Chapter 2).

Overall, the two projects provided quite contrasting observations and experiences for the practice-led researcher in terms of the collaboration practices that occurred with developers. Across both of the projects, it was observed that UX designers and developers were predominantly separated in their organisational divisions and physically sat in different areas of the same floor of their building and they appeared to be a cultural separation and what was frequently mentioned as a "gap" between the two roles.

Each morning the developers would attend stand-ups and throughout the week would have regular retrospectives and sprint-planning meetings – the UX designers did not attend all of these meetings – their involvement would vary due to being involved in multiple projects with different ASD teams so their commitments would change. This separation was addressed through sporadic meetings between the roles that would allow the roles to come

together to discuss their work. However it became clear that this did not address issues or provide the right amount of collaboration between the roles creating a point of friction.

The themes that emerged from the data across the two projects are presented in this section. While these themes centre on the participant-observers UX work across the two projects, some of the themes that emerge are related to the organisational setting and structure placed upon teams and disciplines. The findings for each project are detailed separately below before being discussed in the discussion section (3.5) of this chapter.

3.4.1 Project 1

This project required a series of website feature developments, the researcher and surrounding UX team were involved in different aspects of the project at different times but for two months worked with the ASD team in the product to produce several new features for the site. Overall, it was observed that the UCD and ASD processes were not well integrated and that there was a distinct separation in the organisation's divisions between UX designers and developers with little evidence of close collaboration between the two roles – instead, there were frequent examples of heavy handovers and a lack of communication, causing frustration and inefficiency's between teams. The UX design team were not only separated divisionally but also in location from the ASD teams. The two teams also experienced key decisions being made mostly by stakeholders (team managers) rather than the individuals and the teams being empowered to make decisions themselves. The emergent themes from the data in this project are described in more detail below.

Theme category 1: Work not integrated between roles in the project team

In this project, it was observed that the product team and 'stakeholders' would typically define the work on the project. This was either based on the roadmap backlog or based on problems captured from the audience. The UX designers as part of the design team would work up-front to create solutions for the work that would then be "picked up" by the development team in 2-week sprints.

The proposed UX solution would be thoroughly tested with users and follow typical, well rehearsed UCD methods, usually involving usability testing in a lab environment, or sometimes involving guerrilla testing if changes were smaller with less complexities. Both testing scenarios would often involve a prototype of varying fidelity. One of the key

observations here was that throughout this process, developers would not have been involved. Only once the design process had been "completed" would the ASD team then shape the requirements and the developers would work on it. "Handovers" of static design was often mentioned in relation to finishing UX designs and regular iterations on the product were infrequent.

Often it was the case that the developers would be working on a completely different aspect of the project to the designers, so when the two roles did come together in periodic planning meetings and discussions, there was not the right level of insight and to answer each others questions so often discussions were based around hypotheticals which did not help either role.

This presented what was perceived as a lack of integration between the UCD process and the ASD process and a collaboration gap between designers and developers. Likewise, designers were not regularly part of the ASD process, so there is a lack of or no design representation within typical ASD points such as during sprint planning and retrospectives. This lack of regular dialogue between the roles and lack of collaboration early on in the process created frustration and wasted time. The notion of design work getting "completed" by the design team and delivered to the developers also showed that continuous improvement and iteration was not a key part of the products development process so it did not allow for changes and improvements over time with regular user involvement which are key parts of both the UCD and Agile methodologies.

Theme category 2: Separate teams and locations

The working environment is an open plan office space, with many banks of desks where every individual works out in the open at their workstations. When not at their desks, people are attending meetings within various meeting rooms or breakout areas and individuals, on occasion, are working from home.

The UX design team by organisational design were separated in location based on the "floor plan" in the office and were permanently sat separately to the ASD teams, further exposing the literal and disciplinary separation between the UX designers and the ASD teams. In practice, this separation in seating arrangements was to due to the UX team working across multiple projects at once so requiring them to support different ASD teams.

Despite this, the environment is setup as an open collaborative space so that people can move around, work in break-out-areas and find opportunities to collaborate on an ad-hoc basis. However it was observed that this sort of behaviour did not occur very frequently outside of formal meetings and the silos of the different teams. One of the factors inhibiting this ad-hoc collaboration other than the clear separation in teams is a limited amount of space to collaborate away from somebody's desk, typically there would be no free desks and the open spaces would be noisy and unsuitable for anything other than short meetings and chats. There was also a limited amount of wall space, white boards and areas for off-screen work-in-progress discussion.

There were also perceived benefits of the design team being separated from the ASD teams in that the UCD process could be applied without friction and the team could think holistically about the product with frequent knowledge sharing between UX designers of varying experience and skills. Likewise, with the ASD team located together it permitted for strong collaboration between the developers, business and testing specialists so release deadlines and sprint targets were frequently met.

Theme category 3: Heavy handovers between designers and developers

As discussed, design work in this project occurred mostly upfront without the input of key development insights and technical feasibility and the two roles were typically not working on the same part of the project at same time.

Design work was typically very holistic and worked on at a page (URL) level and was delivered to developers as either a flat mock-up which was created in design software (Adobe photoshop, illustrator or other). Occasionally a prototype would be created with prototyping software (Axure or Proto.io) but often this would be too time consuming and only one or two designers in the team could code or prototype quickly enough for it to fit within the timescales of the project. Regardless of the type of output, designs would be "handed over" to developers for them to then "productionise". This handover would often occur via email following an initial planning discussion involving the ASD team and the relevant designers.

The effect of this upfront design work and handovers of design documentation was observed to cause problems between the two roles and the surrounding product team.

Firstly, at the point of handing over work from a designer to a developer, friction would occur due to an absence of earlier communication and discussion about the vision for the feature or page being built. A developer would also have to laboriously un-pick various aspects of the design to understand how the implementation in the browser would either look or behave during different circumstances. Secondly, development would often occur at a component level rather than at a page level so often some of the more holistic aspects of the design failed to be implemented correctly causing frustration with the UX designers. These problems were particularly apparent for responsive components (layout 'break-points' are used alter the design for different screen sizes) where changes must occur in the UX across numerous different devices sizes.

It was often the case that certain aspects of the proposed UX that was handed over could not be created and would be developed differently to the intentions of the UX team without their knowledge. This was often due to the Agile methods being employed by the development team to release smaller, working releases of the software continuously to the audience rather than all at once which would be more risky. Without common methods for regular dialogue between the roles this would cause regular frustrations as the project progressed resulting in often a lot of "last minute" requests and changes from the UX team. The UX team would then experience frustrations because a lot of time will have been spent producing "spec" documents that then have to be laboriously changed.

Theme category 4: Frustrations with the decision-making process

In this project, it was observed that stakeholders and editorial colleagues who were not directly working closely with the designers and developers would often make decisions. This sign-off process caused frustration and inhibited the speed of the UCD and ASD process due to teams needing to have periodic meetings to get "sign-off" on the work. Whilst there was a desire to improve the decision-making process using audience feedback, data and research at a more localised level in the teams it was not very apparent amongst the day-to-day work in the teams.

Smaller decisions and iterations that did not require a form of sign-off provided the opportunity for roles to make decisions about their work but it was not common for this to be a collaborative exercise between different roles. Again, this appeared to be due to the separation of the different teams in the product so crossover did not occur other than in

meetings and time was often pressured so smaller decisions could not be discussed. As UX designers were working across different projects, they were often needed to be available for quick ad-hoc decisions during the development process.

Theme category 5: Current HCI landscape brings about particular challenges

An emerging theme as the project went on was the challenge presented by trying to design and develop the website for a device of any size as part of implementing what is known as Responsive Web Design (Marcotte, 2010). Prior to this change, designers in the team were used to creating UX designs and wireframes for one size (e.g., 960 pixels in width) to suit a desktop monitor or a single smart phone. However with the advent of responsive design and the huge proliferation of mobile devices of all shapes and sizes, the best practice was to allow for the site to be 'fluid' which helped with catering for multiple devices by not using 'fixed' widths. This meant that designs were created at a few different 'break-point' sizes but then adjusted in the browser depending on where the design layout "broke" and another breakpoint would be added. The challenges this presented was that without tight collaboration between designers and developers, it meant that designers would be tasked with created many more specification files, further adding to their frustrations and a particular challenge once the development began.

3.4.2 Project 2

Prior to conducting reviews of current relevant literature and potentially running more conclusive research studies, *Project 2* offered an opportunity to do more practice-led research and gather further insights from the current team. This project was contrasting to the first project. Due to timescales of a particular piece of work, there was a change in priorities on the product and there was an opportunity for the researcher to work on the UX design of a particular feature alongside a single developer for what turned out to be a period of 4 weeks. Its important to note that this change in workflow was partly dictated by the researcher and surrounding stakeholders because of the challenges of the previous work in *Project 1* that have been discussed.

Overall, in *Project 2,* the design and development roles became much more joined up in both collaboration and communication factors in comparison to *Project 1*. Despite this, the overall integration of UCD and ASD between the wider teams was largely unaffected.

Theme category 1: Early and frequent communication

From the very outset of the project, both roles got together in the same location and spent time communicating and collaborating together to understand the problem. This provided an opportunity to have an open dialogue about the hopes and fears for the project and it levelled expectations and got the work off to what was perceived as a positive start because they both had an understanding of the vision of the project.

Although this felt different to what was typical in the project process, getting together early on also enabled both roles to establish good rapport between each other and set expectations about timings, responsibilities and how they would approach collaborating with each other. As the researcher was the UX designer on the project, it helped to gain understanding of some of the limitations of the data available and to discuss the possibilities of some of initial ideas. Likewise, it helped the developer to understand the impact of the UCD process and how they could share their ideas and incorporate user involvement. This communication impacted early ideation and the initial solutions to the work.

Communicating frequently in this way felt really positive in contrast to the previous project where the design and development work was completely separated. There were times when communication was not necessary, as the roles would need to work independently, but the ad-hoc nature of being next to each other felt beneficial.

Theme category 2: Side by side collaboration

By sitting side-by-side and working together it sped up a lot of the previously observed shortcomings of the previous project. The process was iterative and there was not any handing over of responsibility or a need for documentation. It was efficient and helped to keep the focus on the UX and user involvement as opposed to churning out specifications or 'Acceptance Criteria'.

The usual work tickets that were created and applied in 2-week Sprints as part of the Scrum process were not required nor were the tickets as granular because of the regular communication. This reduced any handover time significantly and meant that parts of the process took less time compared to normal and the work was completed at a faster rate, for example sprint planning for this feature took hardly any time at all.

Sitting side-by-side enabled prototyping really early on in the browser together which proved to be a successful part of determining the usability and possible viability of the work to be carried out too. This meant that when the Acceptance Criteria of the feature was proposed with the wider team it was simple and as a pair we already understand most of the requirements. This was a dramatic shift from the previous project and what was commonly described as the "handover process" in that it was like the roles were working in tandem with each other, frequently pairing to tackle problems rather than being in isolation.

Theme category 3: Knowledge sharing & experimentation

One of the most valuable aspects of coming together to collaborate in this way proved to be the chance to combine their discipline specific understanding and knowledge of the problem at hand. This allowed for in-depth discussions about the technical opportunities and constraints of different ideas whilst always considering the user and the project objectives.

One key aspect of this was discussing and quickly prototyping initial designs in the browser and across different breakpoints. This was particularly valuable when discussing the interaction design of the feature and enabled experimentation of early ideas and to find areas for new and innovative features in the project.

The limitation of this finding was that the researcher had some previous knowledge of front-end development and it helped as a designer to get up to speed. Despite this, it was still a dramatic change from the previous process where knowledge sharing would mostly occur on a need-to-know basis, providing the bare minimum amount of discipline sharing to get the job done. Whilst knowledge sharing like this felt invaluable at the time, it was difficult to measure what was useful to know about and what was not required for each role and whether the increased knowledge sharing and communication negatively impacted upon the project at all.

Theme category 4: Making informed decisions together

Both disciplines were given the authority to make a lot of decisions during the design and the development of the feature, this was partly due to the feature being classed as "nice to have" – meaning that it was not critical to the overall success of the project. Despite this, the feeling of autonomy gave both roles a feeling of empowerment in their work, which in turn enabled quick and iterative decisions. These decisions were grounded in expertise that spanned across

both of the disciplines and from regular user involvement to test designs whilst working iteratively on the features. This gave both roles a greater sense of pride and satisfaction about the direction of the project – knowing that they had worked through countless different directions together to iterate and get to what was thought to best the outcome.

Despite the positives of being able to make decisions quickly and iteratively in this way – it was felt by the wider team that occasionally it was difficult to keep track of what was being worked on and what the direction of the feature was at certain times. This in turn hampered the wider planning of the two weeks sprints and the work of the other team members.

Theme category 5: A more robust and confident solution

When the work had been completed and the feature released to the audience, pride and satisfaction was felt between the designer, the developer, the researcher and the rest of the team. This came from the confidence that what had been produced was of a high quality and was as both roles intended it to be. This included the consistency of the interactions and visuals across devices and their confidence in usability, which had been tested rigorously throughout the process.

At this stage, key stakeholders and other colleagues also commended the efficiency and output of the UX design. The appreciation of the feature by the user was more difficult to ground and after the initial release there was an opportunity to learn, gather feedback and then further iterate on the feature to make it better for the user while it was being used.

Theme category 6: Lack of wider sharing and awareness

Despite the initial success of coming together in this way on the project, the UX designer had been so focused in the project that there hadn't been as much time to think holistically about other parts of the project with the other UX designers. So whilst the collaboration with the developer had improved – working closely with other members of their teams was reduced. However, the project was relatively short and once the initial release had occurred, the UX designer and developer could focus on some of the other work but it was important to observe this effect on the wider team and projects.

3.5 Discussion

The preliminary practice-led study during *Project 1 & 2* provides an initial exploration into different aspects of UCD and ASD and their integration in a large organisation. In particular, this study provides an insight into the factors that affect collaboration between UX designers and developers.

Project 1 explored how people work together in an organisational setting and because of the researchers working practice as a UX designer, the study focused in on how UX designers and developers work and collaborate together in an organisation that typically adopts an ASD process. The insights gathered from this part of the study showed there was a separation between the UCD and ASD processes which was emphasised by a lack of collaboration between designers and developers. Coming together to collaborate only appeared to occur on a need-to-know basis and often half way through a project. This lack of integration in project work made the roles frustrated and caused an adverse effect on the relationship between the two roles.

Little interaction between the two roles took place and typically a large documentation based handover was involved and the UCD process occurred without any regular contribution from developers. A common phase of "designs passed over the wall" was mentioned, contradicting the ASD value of aiming to reduce comprehensive documentation. The UCD processes of involving users and thinking holistically was thoroughly adopted by the UX designers but without contribution from the developers. This caused a lack of knowledge sharing and a separated workflow, resulting in a lot of documentation. Frustration would arise from both roles in this culture of separation and handovers; Designers would have frustrations about how their UX vision had been created by the developers in the browser and developers would have frustrations because of a lack of involvement earlier on in the design process. Motivation and individual empowerment was also found to be lower by the lack of shared decision-making between designers and developers and the go-to outcome was to defer decisions to stakeholders for "sign-off".

These findings from *Project 1* ultimately adversely affected the efficiency and enjoyment of the work due to a lack of collaboration and communication early on in the project and an absence localised decision-making. Developers would spend lots of time having to unpick designs, and designers would spend time trying to create things that could not be built in the

browser, causing mutual frustrations. Despite the problems and lack of integration that had been observed by the researcher, there was an opportunity to improve upon the collaboration between the two roles. *Project 2* provided the chance for the researcher to start working more closely with a single developer. From the perspective of the research study, and because of the insights gathered during *Project 1* the aim of this project was to begin to understand the effect on a project when a UX designer and developer start to collaborate from the outset. The aim in particular was to find out if this collaboration technique would address some of the problems and challenges that the first project identified around handovers, communication barriers, location and decision-making.

Interestingly, the effect of coming together in *Project 2* to collaborate was quite contrasting in comparison to the first project, the two roles showed a pairing-like behaviour to collaboration where they frequently worked side-by-side and communicated regularly throughout the project. Because of this, their knowledge sharing increased, especially early on in the project. This informed how the work was tackled which was largely seen as a positive effect.

Iteration is a key aspect of both UCD and ASD principles, but it was not a frequent aspect of the design and development process in *Project 1* and this was much improved in *Project 2*. Iteration was improved by working quickly alongside one another, making decisions and solving problems together – this was particularly evident from their work in the browser – to collaboratively fine tune the UX and make prototypes together. Although the decision-making became more frequent between the designer and developer in *Project 2*, the change had an adverse effect on the awareness of the project from the wider team. By being so joined-up it limited the amount of time the designer and developer could work with other colleagues whilst the project was on going. However, it was perceived that because the project had a finite lifespan, other work with colleagues could be picked upon its conclusion.

Despite this study only concerning observations about the researcher and their own surrounding team, in *Project 2*, there were dramatic changes in outcomes from what had been observed previously. The adjustment of the UX designer from working in a separate UX team to working directly with the developer from the outset proved to help in integrating the two roles together and had seen profound changes to their workflow. The developer and the

surrounding colleagues regarded the collaboration techniques as a success, especially in reducing handovers and increasing communication early on in the project.

3.6 Summary

This chapter presented an ethnographically informed study of an established product team in a large organisation. The preliminary practice-led projects described in this chapter provide an understanding of the state of collaboration between designers and developers in the context of the researchers own organisation, their role, the team and their practice. Through practice-led research, this study identified what appeared to be a problem area and an opportunity to understand and develop collaboration methods and practices between designers and developers through further, more empirical based study.

The overall findings from *Project 1* showed that the UCD and ASD processes were at odds with each other and specifically there was little evidence of regular collaboration or communication between UX designers and developers. Both projects did however show an observed success in respective areas of UCD and ASD but for separate reasons. *Project 1* demonstrated that although UCD and ASD were at odds with each other, the design team could work in isolation and think holistically about the design and research on the projects, aligning with one of the key principles of UCD. The developers on the other hand could work through many aspects of the projects iteratively and without friction where different ASD methods (e.g., two weeks sprints) would work well. The challenge was when the two roles needed to come together to create the UX for the features or projects.

In particular, *Project 1* exposed a range of observed problems between UX designers and developers that could be more thoroughly investigated in literature and across a wider set of teams and projects to identify and understand if these issues are commonplace or just isolated to this particular scenario and team.

The theory that some of these factors could be significantly improved by bringing a UX designer and developer together was initially examined within the existing team for *Project 2*. This showed success particularly around side-by-side communication, iterative decision-making and idea sharing early on in a project. The two roles adopted a paired-like approach where they worked so closely together that the lines between the two disciplines became

blurred. This approach also came with its challenges in that they worked so closely together some of the other team members felt excluded and it was more difficult to keep track of on going changes and iterations.

3.6.1 Next steps

With the researcher participating in the research it does not mean that it is of less rigour because it is important to keep the subjectivity in check and present and analyse the evidence objectively. Also as identified previously and in the research design (Chapter 4), acting as the participant observer within the organisation provides a unique perspective that an external researcher would find difficult to capture. Despite this, the study was only preliminary in scope and was restricted to the researchers immediate team. Prior to and alongside this study, the current literature (Chapter 2) surrounding the integration of UCD and ASD and the reported sources of how UX designers and developers come together to collaborate has been investigated. The next step is to expand the research further to a broader, more comprehensive range of teams and individuals at scale, and where the researcher is abstracted as a participant.

Chapter 4
Research Design

4.1 Introduction

This chapter provides an overview of the methodology for the research that was carried out. The main focus of the research in order to answer the key research questions was to investigate industry practice in a complex organisational setting. The research was emergent in nature, meaning that it evolved as the research was carried out and was informed by the review of the reported literature (Chapter 2) and by the practice-led preliminary study that was carried out (Chapter 3).

The next sections in this chapter describe the research background relevant for this research (4.2), followed by describing the selected research methodology for this study (4.3) and finally a summary is then presented (4.5).

4.2 Background

This section discuses the research design background that underpins the research methodology that is discussed in section 4.3.

4.2.1 Designing the research appropriately

This relationship between the research questions and the research design is fundamental to the whole research process. If an unsuitable research design has been used to answer a particular question, the quality of the research itself will be fundamentally challenged. So it is important that the research question is matched with a suitable design (Closs & Cheater, 1999). And to achieve this it is important to make sure that the research approach aims to answer the main research question (Draper, 2004).

Designing the research question(s)

The type of knowledge the researcher intends to generate often determines the development of the main research question(s). This might be selected as descriptive, explanatory or predictive knowledge or a combination of the three. Research that generates descriptive knowledge tends to be when very little is known about a topic and will often involve exploratory qualitative methods. If the research intends to further explore a relationship between different components of a topic that have already been researched, then explanatory knowledge will be generated. Lastly, if descriptive and exploratory knowledge about a topic is already available, the desire might be to generate predictive types of knowledge. Predictive knowledge is concerned with confirming cause and effect relationships (e.g., X will/will not have Y affect on Z) and experimental research methods with controlled variables are often used to answer such questions (Iacono, Brown, & Holtham, 2009).

Some research questions are so specific that only one type of research design will be appropriate, whilst others will be more ambiguous and may be informed as the research is undertaken. When designing the study it is important to consider the experience of the researcher and any other support or resources that might be needed and implicate the complexity or ethical considerations of the research.

Paradigms to inquiry

Based on the research question and the type of knowledge that intends to be generated, the research will be within a particular research paradigm. This could be a paradigm of positivism, in that the view of the world to the researcher is 'objective' and everything can be captured and measured to prove theory and establish a cause and effect, resulting in predictive knowledge, this type of research often involves quantitative research methods. Or, this could be using a naturalistic paradigm. This type of paradigm assumes there is not necessarily an objective reality or truth and because we are all different, a number of realities can exist. Research within a naturalistic paradigm is concerned with understanding these different perceptions and taking into account the cultural context in which the research takes place where the position and effect of the researcher is considered an important factor. Within this paradigm, types of descriptive and explanatory knowledge are generated, often using qualitative research methods (Draper, 2004).

In this research, the inquiry is based on a naturalistic paradigm in that the reality as a multiple, constructed, interdependent whole is not easily reduced to numbers (Bailey, 1997). This is due to the research being focused on how people behave in a social context and when absorbed in their work as a genuine life experience in a natural setting. One of the types of naturalistic inquiry which is used for this research is ethnographic – which uses direct observation and field work to produce a naturalistic description of people and their culture (Frey & Botan, 1999).

4.2.2 Collecting data in a naturalistic inquiry

There are two main strategies that are used in the context of naturalistic research to collect data, these are through observing and interviewing methods. As discussed, this type of research involves going "into the field" to undertake fieldwork to explore everyday activities. In this research it is important to consider the role of the researcher in the fieldwork and one of the challenges within this study comes from the researcher being employed as a UX designer as part of the organisation, the setting for the research itself. This presents a unique challenge as well as a unique opportunity for the research.

HCI & SE research approaches

In HCI and SE, there are many complex social based phenomena that cannot easily be quantified or experimented upon in research. For example, as described by Adams et al (2008), the role of privacy in HCI is not obviously reduced to a number and it would not be appropriate to limit a person's privacy in the name of research (Adams, Lunt, & Cairns, 2008).

In addition, technology is rapidly changing – the proliferation of mobile devices alone makes it harder to abstract technology from the context of use to study it effectively. Instead, complex interactions between technologies and ways of life are coming to the fore. Consequently, it is common that the real issues of the problem are unknown before the research has begun. This makes it difficult to define the variables necessary to do solely quantitative research.

Moreover, the differences between qualitative based research data and quantitative research data have seemed of huge significance to many social researchers (Bryman, 2004). Despite this, the type of data is no longer thought of as a determining factor of selecting a particular

research method. Therefore a mixed method approach that combines both quantitative and qualitative data and uses multiple research methods is now fully accepted as a valid approach to a HCI & SE enquiry (Cavaye, 1996; Myers, 1999).

Interestingly as described by Latour (1987), who argues that there is basically little difference between the two paradigms, as both qualitative and quantitative endeavour to arrange and rearrange the intricacies of raw data (Adams et al., 2008). Moreover, (Bryman, 2004) additionally suggests that the distinction between these two approaches is purely technical, so that the choice between them relies on their suitability in answering particular research questions. In the case of this research both qualitative and quantitative methods are useful bearing in mind the naturalistic context of the research and the lack of current quantitative studies within this particular area. It is suggested that using a combination of methods like this is a valuable approach towards strengthening the quality of research (Henwood & Pidgeon, 1992).

Participant based observation in an organisational setting

Since organisations can be viewed as societies with their own peculiar customs and practices, participant observation has become popular in organisational research. Participant observation is described by Iacono, Brown and Holtham (2009) as having its roots in anthropological studies, where researchers would travel to faraway places to study the customs and practices of less known societies. It involves participating in a situation, while, at the same time, recording what is being observed (Iacono et al., 2009). Hence, participant observation has been associated with qualitative methods, as the data collected by this technique tend to be predominantly qualitative. It is potentially rewarding but presents unique challenges to the researcher. It provides the chance to gain unique insights into an organisation or social group.

A criticism levelled at participant observation is the potential lack of objectivity, as the researcher is not an independent observer and the phenomenon being observed is the subject of research. The notion of participant observer does presuppose a degree of emotional detachment from the subject matter, the clear objective of the researcher being the conduct of the research. American Liza Dalby moved to Japan and lived as a geisha among geishas to conduct the fieldwork for her PhD thesis, and later recounted her experience in the field in the book ‘Geisha’ (Dalby, 2000). As a researcher, she faced a similar challenge, namely, how

to reconcile her very personal experience and views with the need for detachment traditionally expected of a researcher for her work to be regarded as scientific.

Inevitably participant observation raises ethical dilemmas: the investigation should not be conducted in a covert manner; participants should be informed of the nature and scope of the investigation. On the other hand, participant observation carries with it the concern that the presence of the researcher may influence the way participants behave. Informants may be suspicious of the researcher and reluctant to participate or be eager to please; they may interject their own impressions and biases etc. The personal relationship between researcher and informants may also influence the interaction (e.g. the researcher may empathise with participants and vice versa).

This type of reactivity is known as the *Hawthorne Effect*, or observer effect, which was a term coined by Henry Landsberger in 1958. Workers at an electric factory in America were observed to become more productive when the study commenced but productivity slumped when the study into the lighting effects ended. Suggesting that productivity improved as a result of a motivational effect because interest was being shown in the workers (Landsberger, 1958). Both outside and inside research have pros and cons. The former may be methodologically precise, but yield results epistemologically irrelevant; the latter, on the other hand, may appear to lack rigour and/or objectivity.

Within this study, these shortcomings may be mitigated to a certain extent by alternating between the two modes and there will be the opportunity to do this by working across the organisation and combining both qualitative and quantitative research methods.

4.3 Research methodology

This section describes the overall research methodology that was designed for this study, including the approach and the main events for each study along with the ethical considerations that were met.

Throughout this study, the research design was emergent in that it had "the potential to change or shift emphasis" (Barbour & Barbour, 2003) as the research was undertaken and

data was gathered. The findings could influence the next steps through the analysis and could determine the focus for the following research methods.

As discussed above, defining the research question is an important part of the research design (Yin, 2003). For this study, the broad research question was *How do UX designers and software developers work together to create online products and services?* The selected approach for the research depends on the field(s) in question on how it can be understood. This research focuses on the fields of HCI & SE and the emphasis is on understanding and informing practice in the design and development of software. To do this, the research should be focused on people as practitioners and on the settings in which they work (Lethbridge, Sim, & Singer, 2005).

All of the research undertaken in this study took place in the context of the researchers organisation. This is a large organisation based in the United Kingdom, it includes several very large online products with an output of editorially steered content alongside traditional TV and radio broadcasting. It is structured into audience facing product teams, each with their own set of UX designers and software developers.

To answer this first research question, an ethnographically informed field study was conducted and explored how UX designers and software developers work together to create online products and services. This study was practice-led by the researcher working as participant observer and took place following a review of the current literature.

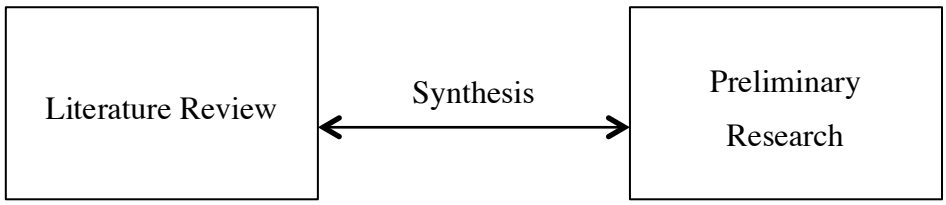

Figure 4-1: Initial Research Design

Following a review of the literature and the analysis of the preliminary study, the results were synthesised together (shown in Figure 4-1) and the research question(s) were refined to *How do people integrate UCD with ASD in practice and what is the impact on collaboration between designers and developers in an organisational setting?* – To achieve the aims of the

naturalistic inquiry, a mixed-method approach was selected, using a combination of surveying and in-depth interviews as sources of data to answer the question.

Following the analysis of the in-depth interviews and survey, the emerging themes from the research refined the research question(s) further to *What does 'success' look like between pairs of designers and developers working in an UCD and ASD environment? What are barriers and what are consequences of successful paired work?* – To answer this, fieldwork was undertaken over the course of two separate studies, both in the form of a 'pairing study' with UX designers and software developers. The first obtained data via surveying and the second used a Contextual Inquiry and Diary Study. The contributions presented in this thesis are a synthesis of the results of the mixed method approach to the research design and an overview of this is shown in the figure below.

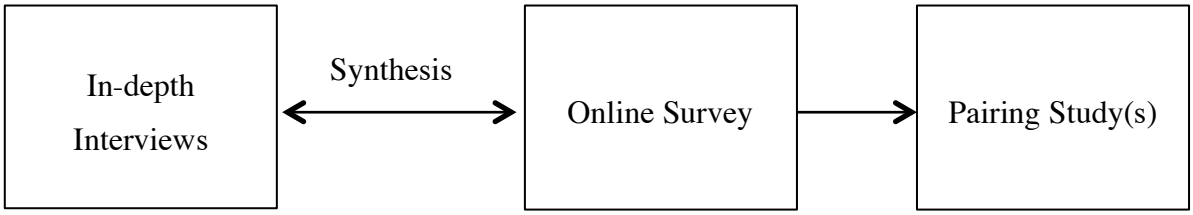

Figure 4-2: Mixed Method Research Design

4.4 Study design

This section describes the role of the researcher, the participants, the data collection and finally the analysis methods that was used for each study. More specific details for each study is described within each study chapter (Chapters 5 – 8) and the following describes the main events for each and how each was informed by the findings from the former analysis.

4.4.1 The role of the researcher

The role of the researcher varied as the different studies were carried out but broadly it can be described as an 'informed observer', this included a combination of complete participation in the preliminary study then changing to the role of a complete observer in the other fieldwork studies.

4.4.2 The organisational setting

Opportunely, in conjunction with the research aims, the large, complex organisation of the researcher permitted that data collection could take place across the organisation to drive this research forward (reference to this authorisation is noted in Appendix A). Thus, all of research studies took place within the context of a large media and broadcasting organisation based in the United Kingdom, with ca. 18,000 employees over all.

The organisation is particularly complex as it is based across a variety of locations and includes several very different online products with an output of editorially steered content alongside traditional TV and radio broadcasting. The organisation's remit is to inform, educate and entertain everybody in the UK and alongside TV and radio content has a Design & Engineering department which creates apps, website and digital services for its different products and brands. The particularly complex nature of the organisation, featuring many brands, products and locations, makes for an interesting research setting that will likely extend findings to other complex organisations as factors considering in this setting are unlikely to be unique.

4.4.3 Participants

The UX designers and software developers in the organisation work alongside Business Analysts, Testing Specialists, Project Managers and Architects, all with varying experience and responsibilities. The disciplines within the teams report to a set of product "stakeholders" – a term given to managerial staff - that includes product managers, technical leads and creative directors. Designers, developers and their stakeholders were used as participants in the research and were all from within the organisation. For the in-depth interviews in *Study 1*, 9 stakeholders of varying roles were interviewed and in *Study 2*, the online survey, there were 109 participants, (24 women) in the sample and 54 were classed as designers and 55 as developers. In the pairing studies, 2 participants were used in *Study 3* and 12 participants were used in *Study 4*, each with an even split of UX designers and software developers.

To select participants for the in-depth interviews and the online survey a sampling method known as 'purposive sampling' (Palys, 2008) was chosen. This enabled the studies to be focused on the right roles within in the organisation, UX designers, software developers and their stakeholders. To gain suitable participation and the appropriate amount of data for the

study, people from across the organisation were invited by email to take part. All of the product teams that took part in the study were reported to have, by design, adopted the ASD process to varying degrees of flexibility; some involving mixed approaches to Scrum and Kanban methods.

4.4.4 Data Collection Methods

This section discusses the data collection methods that were used during the research studies. As discussed previously, a mixed method research approach was selected to answer the questions outlined in the research. This resulted in different data collection methods that were used for each of the main studies. The type of data that was collected is outlined in Table 4—1 below and more specifics about the particular measures are detailed in the respective study chapters.

	Preliminary research	**Study 1 - Stakeholder interviews**	**Study 2 - Online survey**	**Study 3- Pairing Study A**	**Study 4 - Pairing Study B**
Method	Practice-led observations	In-depth Interviews	Online questionnaire	Online questionnaire	Contextual Inquiry and Diary study
Data Collected	Field notes	Transcriptions and notes	Raw data of both the ratings questions and open responses	Open responses	Transcriptions, field notes and diary entries

Table 4—1: Data collected for each study

In-depth interviews

An in-depth interview is a qualitative interview technique that aims to capture insights and perspectives about a particular idea, program, culture or scenario. It involves conducting thorough individual interviews with a small group of participants where open-ended questions are asked to elicit depth of information. When successful, an in-depth interview deeply explores "the respondent's feelings and perspectives" (Guion, 2006, p. 1) on a particular subject.

In-depth interviews are particularly useful when detailed information is required to gain the full picture of a situation or an event. Their primary advantage is that "they provide much

more detailed information than what is available through other data collection methods, such as surveys" (Boyce & Neale, 2006, p. 3). This technique is also useful for obtaining information to later inform and refine survey questions of a particular group or topic. As part of an in-depth interview its important to ask open-ended questions as part of what feels like a natural and fluid conversation with the respondent. In Study 1, the aim was to obtain a deep understanding about the integration of ASD and UCD and collaboration between designers and developers across the organisation. The in-depth interview technique was well suited for these aims and an interview protocol was developed for the study in the form of a guide, which included details for each interview session and an accompanying script.

The interview script was created for Study 1 to reflect and try to answer the main research question for the study. As outlined in the section above, it is important to outline a script that guides the conversation with the respondent and allows for flexibility based on their response and situation. The style of an in-depth interview is informal, rather than ticking every box in an interview script but instead aim to guide the flow of the conversation with the participant(s) - detailed further in the study (Chapter 5).

Questionnaires

Questionnaires as a research method, whether using online, digital or paper-based methods are one of most widely used means of collecting data (Rowley, 2014). They are typically used in survey situations, where the purpose is to collect data from a relatively large number of people. Questionnaires provide both quantitative and qualitative methods of gathering data and offer an objective means of collecting information about people's knowledge, beliefs, attitudes, and behaviour (Oppenheim, 1994; Sapsford, 2007). For this research the questionnaire method was used for Study 2 and 3 to gain a broad understanding of participants perspectives and to answer the particular research questions for each study. The questionnaires were designed using a combination of closed and open questions, aiming to leverage the appropriate insights to help inform the overall study.

For Study 2, an online questionnaire was designed to capture insights from a large number of participants across a large organisation. The online questionnaire was designed to ask designers and developers about their roles, perceived level of ASD implementation, satisfaction with ASD, teamwork satisfaction, and perceived quality of communication and collaboration with the other role. In particular, the study aimed to explore whether designers

and developers had different perspectives in terms of how ASD works in their organisation, how well the wider team (including other roles, e.g. business analyst) worked together, and whether there were specific issues in designer- developer collaboration and communication that would impact on the current processes, their goals or objectives. For the open questions in questionnaire, questions invited respondents to provide their perspectives as comments (typically between one and three sentences). For the closed questions, the Likert scale (a scale developed by Likert (1932) and used to represent people's attitudes, opinions and beliefs about a topic) was used to capture respondents ratings. Study 3 invited a design and development pair to fill out an online questionnaire at the beginning and at the end of the study. This online questionnaire was designing using only open questions and aimed to capture salient comments about the participant's experiences during the study.

Contextual Inquiry

A Contextual Inquiry is a qualitative data-gathering and data-analysis methodology adapted from the fields of psychology, anthropology, and sociology (Darroch and Silvers, 1982, Glaser and Strauss, 1967). As reported by Raven (1996), a Contextual Inquiry essentially consists of observing and talking with users in their workplaces as they do real work.

A Contextual Inquiry (CI) is a semi-structured interview method used to obtain rich information about work practices, the social, technical, and physical environments, and user tools (Wixon, Holtzblatt, & Knox, 1990). Participants are first asked a set of pre-defined questions and then observed and questioned while they work in their own environments. This method that can be adapted to suit different situations and goals. The interviews and observations take place in the participant's normal working environment where they can explain their typical processes, tasks, environment, opinions, tools and their communication patterns. A CI defines four principles to guide the interaction:

Context—Interviews are conducted in the participant's actual workplace. The researcher observes participants do their own work tasks and discusses any artefacts they generate or use with them. In addition, the researcher gathers detailed stories of specific past events relevant to the project focus. If specific tasks are important, the user may be asked to perform those tasks.

Partnership—Participant and researcher collaborate to understand the participant's work. The interview alternates between observing the participant as they work and discussing what they did and why.

Interpretation—The researcher shares their interpretations and insights with the participant during the interview. The participant may expand or correct the researcher's understanding.

Focus—The researcher steers the interaction towards topics, which are relevant to the project scope.

Importantly, for Study 4 the CI method was well suited. Benefitting from participants being interviewed in their own environments, doing their typical everyday tasks and permitting the capture of more salient, nuanced behaviour and work-environment relationships than standard interviews or user tests. Further details about the particular interview techniques, measures and scripting for the CI can be found in the individual study chapter (Chapter 8).

Diary study

A diary study is a method that aims to collect data on a daily basis or even several times a day. During the past decade, diary methods have been increasingly used in work and organisational research (Eerde, Totterdell, & Holman, 2005), particularly in the areas of emotions at work (Bono, Foldes, Vinson, & Muros, 2007), work – home interface (Butler, Grzywacz, Bass, & Linney, 2005; Ilies, Schwind, & Heller, 2007), and social interactions (Tschan, Rochat, & Zapf, 2005). Ohly (2010) reports that diary methods allow work and organisational researchers to study thoughts, feelings, and behaviours within the natural work context as well as characteristics of the work situation that might fluctuate on a daily basis. Diary methods are useful for capturing the short-term dynamics and perceptions of experiences within and between individuals in the work context.

By using the diary study method, Ebner-Priemer & Kubiak (2007) report that phenomena and processes can be discovered in their natural settings in contrast to a laboratory setting (Ohly et al., 2010). Therefore, diary studies have the potential to provide researchers with an opportunity to capture "life as it is lived" (Bolger, Davis, & Rafaeli, 2002) and appeared to be well suited as a way of capturing insights from pairs of designers and developers as part of Study 4.

To make this work effectively it was important to design the right format for the study and the participants in question. Diary studies usually have an open format where participants record events, thoughts, feelings, and behaviours using their own words (Poppleton, Briner, & Kiefer, 2008). Because of this, participating in a diary study requires more effort from respondents than participating in a traditional survey because of the multiple assessments (Hektner, Schmidt, & Csikszentmihalyi, 2007). Therefore, Reis and Gable (2000) recommended that daily assessments should not exceed 5–7 min in total. With this in mind the setup, instructions and prompts for the diary study are an important part of the procedure and are detailed further in the Study 4 (Chapter 8).

4.4.5 Analysis

This section discusses the analysis methods that were used during the research studies. As described above, the data was captured using both qualitative and quantitative methods and the methods for analysis are described below.

Qualitative analysis

The qualitative data from the data collection methods across the studies was analysed using a thematic analysis approach and was data driven, in that the analysis did not involve any predefined coding scheme or hypothesis. Thematic analysis is "a method for identifying, analysing and reporting patterns (themes) within data" (Braun & Clarke, 2006). In the analysis, themes are identified because they capture something important, patterned or significant in the data. The details of the data collected in each study and the subsequent analysis is discussed in the individual study chapters (Chapters 3, 5, 6, 7 and 8). The thematic analysis across each study was an iterative process that informed the resulting studies and can be summarised in three phases. Step 1 is to familiarise with the data, transcribe, read and re-read it to understand the breadth of the content. Step 2 is to generate codes across the data that imply meaning or significance to the research questions, with codes then applied at different levels to begin to sort the important aspects of the data. Step 3 is to collate the codes and extracts into themes that can then be discussed as findings in context of the other research and the research aims. An example of the coding process from the thematic analysis can be seen below in Figure 4-3.

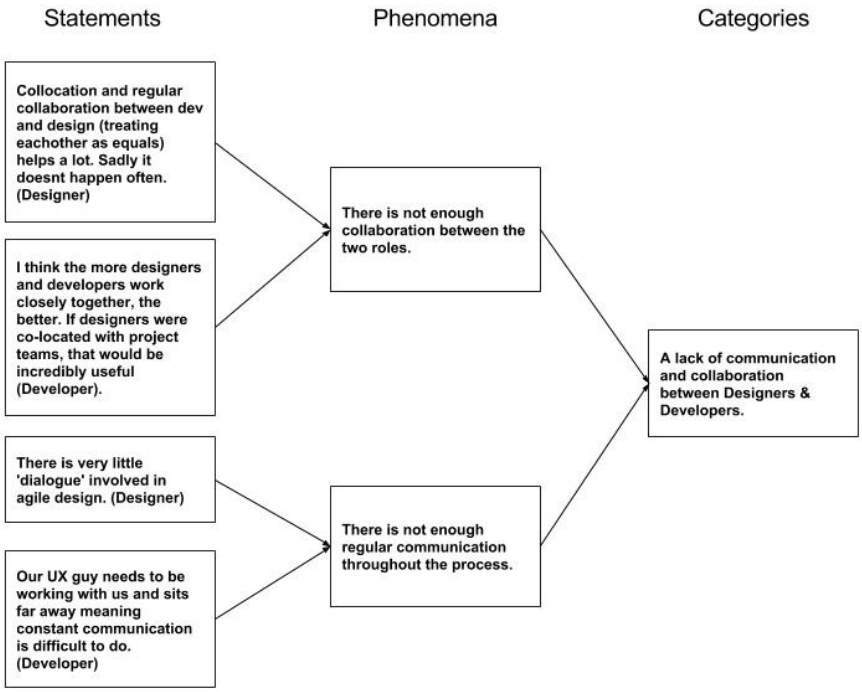

Figure 4-3 - An example of the coding process during the thematic analysis

Quantitative analysis

The quantitative analysis was applied to the rating scores (raw scores ranging from 1 to 7) of the online survey in Study 2. This covered the 3 main sections of the questionnaire – about satisfaction, teamwork, and collaboration. A multivariate approach to the analysis was selected and was undertaken by an external quantitative researcher. This, along with the lead researcher's qualitative analysis of the survey data has been published and contributes to the qualitative findings seen in Study 2 – see Jones, Thoma & Newell (2016) and Appendix B. More details about this collaborative approach to the study analysis are provided in Chapter 6 (specifically section 6.3).

4.4.6 Ethical Considerations

Ethics approval was obtained from the University of Chester's ethics committee for carrying out research of this kind with human participants. Confirmation and details of the ethical approach is provided in Appendix A.

4.5 Summary

This chapter has introduced the background and the details of the research design and methodology used to answer the different questions in this research. The research consists of a practice-led preliminary study and four empirical studies across different teams in an organisational setting featuring UX designers, software developers and their stakeholders. Qualitative and quantitative methods were employed to capture the data that was then analysed to generate findings. The findings across the different studies were then discussed and compared to each other in respect of the research questions in Chapter 9.

While this chapter has provided an overarching understanding of how the research was conducted, the particular details of each research study are provided in the individual study chapters (3, 5, 6, 7 & 8).

Chapter 5
Study 1 – Stakeholder Interviews

5.1 Introduction

This chapter presents an ethnographically informed study of 9 stakeholders (managers) of UX designers and software developers from different teams across a large organisation. The stakeholders were interviewed for approximately 1 hour each using the in-depth interview approach and the interviews took place over the course of 1 week.

The next section (5.2) presents the background information on the participants, the products they work in and the organisational setting. Next, the data collection and analysis methods are discussed (5.3). The main themes that emerged from the in-depth interviews are then detailed: Experiences of ASD methods (5.4.1), Separation between UCD & ASD (5.4.2), Challenges in decision-making (5.4.3) and Desire for pairing and close collaboration (5.4.4). These findings are then discussed (5.5) and finally the key findings of the study are then summarised (5.6).

5.2 Background

The 9 stakeholders interviewed in this study were part of a large media organisation based in the UK. The organisation is detailed in the Research Design chapter (4.4). The stakeholders were line managers of UX designers and software developers from across 9 different product teams in the organisation.

The products were focused into creating websites and apps based on editorial steered content. At the time of the research all of the stakeholders were working on different products involving projects at different stages of product design and development. They were based between the locations of Manchester and London where their respective product team(s) were based.

The aim of this study was to find out how these senior stakeholders perceive the work of their teams and individuals, how they collaborate and what their pain points were. By interviewing people from across the organisation rather than in just one product it helped to gain an understanding across a wide and diverse organisation that featured numerous different projects, team structures and types of roles. In particular the study aimed to answer the following research question that was defined in the Research Design chapter of this thesis:

How do people integrate UCD with ASD in practice and what is the impact on collaboration between designers and developers in an organisational setting?

To answer this question the in-depth interview method was selected for the stakeholders to learn about how they perceive how their teams and individuals work together within the UCD and ASD processes. The focus of the interviews with these experienced individuals was to analyse what had and hadn't worked in the past. The interviews needed to be completely confidential in order to find out about everything that could be undermining the likelihood of successful collaboration.

5.2.1 Participants

The 9 stakeholders were a combination of Creative Directors, Product Owners and Technical Leads from across the organisation and to select the participants for the study a sampling method known as purposive sampling was used. By using the directories of employees within the organisation different senior manager-level participants could be selected by the researcher and contacted for participation. In the invitation to the study, it was indicated to the participants that we wanted to get people's experiences and views on the structure, culture, and processes within the design and development teams and projects, as well as challenges and good practices. Table 5—1 below lists the stakeholder participants that took part in the study:

Role	Individuals
Creative Director	3
Technical Lead	3
Product Owner	3

Table 5—1: In-depth interview participants and their roles

5.3 Data Collection & Analysis

The 9 participants were in-depth interviewed in their normal working environment and the interviews took place in typical breakout areas where normal chats and discussions would often take place. Each interview lasted for about 1 hour and they were recorded and then later transcribed for analysis.

An in-depth interview is a qualitative interview technique that aims to capture insights and perspectives about a particular idea, program, culture or scenario. It involves conducting thorough individual interviews with a small group of participants where open-ended questions are asked to elicit depth of information. When successful, an in-depth interview deeply explores "the respondent's feelings and perspectives" (Guion, 2006, p. 1) on a particular subject.

In-depth interviews are particularly useful when detailed information is required to gain the full picture of a situation or an event. Their primary advantage is that "they provide much more detailed information than what is available through other data collection methods, such as surveys" (Boyce & Neale, 2006, p. 3). This technique is also useful for obtaining information to later inform and refine survey questions of a particular group or topic.
As part of an in-depth interview its important to ask open-ended questions as part of what feels like a natural and fluid conversation with the respondent. To do this an interview protocol was developed in the form of a guide, which included details for each interview session and an accompanying script.

The interview script was created to reflect and try to answer the main research question for the study. As outlined in the section above, it is important to outline a script that guides the conversation with the respondent and allows for flexibility based on their response and situation. The style of the interview was informal, rather than ticking every box in the interview script, the subject focus area was broadly covered with the following questions and topics that are detailed below in Figure 5-1. Importantly, these questions did not aim to act as an exhaustive script, but instead were there to guide the flow of the conversation with the participant(s).

In-depth Interview – Interview Discussion Guide

Setup

- Hello and thanks
- Names
- Purpose
- Confidentiality
- Duration
- How interview will be conducted
- Opportunity for questions
- Signature of consent

Questions about product vision & strategy

- What is your role/position? And before this?
- How does it link in with digital development projects/teams?
- How would you describe the nature of your responsibility for delivering to the audience?
- What are the last products you were involved in?
- In a few words, what describes the organisations digital proposition?

Questions about organisation / proposition

- What is the core value property of the offerings?
- What problems/desires do users have?
- What defines success regarding the user experience?

Questions about different roles & teams

- How large are the ASD teams?
- What is involved in the role of a designer and developer?
- What do you think is their vision for their product?
- Who and how are decisions on product developments / sign-offs made?
- What skills do they have?
- How would you describe the collaboration in teams?
- How would you describe the style of communication in teams?
- What defines success for teams and projects?
- What problems do teams and roles have?
- How could these be solved?

Questions about process, workflow, context and environment

- What technology decisions have already been made, and what's driving them?
- How do teams engage in Agile/Lean UX process?
- What frustrations / pain points do they experience with their current process
- What tools do teams use today? Where do those tools fit?
- What data points do they collect today? What insight are they able to share with each other? How do they share it? What does it look like?
- What's missing in the current process? Why?
- How do you imagine a new offering (process, workflow) will fit?
- What about physical structures? Rooms, offices, and working across different locations?

	Questions about senior stakeholders • What do you think are the main differences between roles regarding the questions above? • What challenges do you see and anticipate for different team member roles? • What are future technological or societal trends that will affect the team processes? • What is your vision for the work processes in teams? How would this fit within the organization?
Closing questions & statements • Additional comments • Next steps • Thanks	• Is there anything you would like to add? • From your answers I will be transcribing what I have and then analysing this with other responses to form into a report and I will be happy to share that with you.

Figure 5-1: In-depth Interview - Discussion Guide

All of the in-depth interviews were recorded and then were analysed using a qualitative approach. Shortly after the interviews took place the recordings were transcribed and the responses were read thoroughly. The transcripts were imported into a software tool for analysing qualitative data called NVivo[5]. Expressions of opinions, problems, events, reactions and interactions in the text were coded by assigning the piece of text to a category (''node'' in NVivo). A category represents a phenomenon, that is, a problem, an issue or an event that is defined as being significant. When categories were found to be conceptually similar in nature they were grouped under more abstract, higher-order themes. Finally, NVivo was used to create connections between themes and their subcategories, which are listed below.

[5] www.qsrinternational.com

5.4 Findings

5.4.1 Experiences of ASD methods

The in-depth interviews found that the respondents felt ASD methods were largely beneficial and helped with the development the organisations products and services. However, due to the nature and context of the organisation it was perceived that Agile methods were not always fully implemented and there was a desire for more a more efficient process which was more iterative.

Theme category 1: Positivity of using ASD methods

(i) Working in an Agile way and using ASD methods (e.g., Kanban) to break down tasks and to release iteratively to users was seen by all respondents as beneficial. Across the organisation ASD techniques are used to varying degrees of flexibility too, which was appreciated by the stakeholders so that their teams could be managed differently depending on the needs of the project and the requirements.

> Agile / Iterative means speed of change and reacting to change of requirements. I would be surprised if anybody could come up with a better process.
>
> The overall goal should be delivering to audience that meets their needs.
>
> Some teams work in Kanban, some in Scrum! With teams we have dependencies on we agreed to communicate as much as possible!

(ii) Although using ASD methods was perceived as being beneficial, the respondents talked frequently about an inconsistency in approaches across the organisation within different teams.

> We (Agile team) work in a very lean way - whereas the rest of the larger organisation (senior management) doesn't want to work in this way!
>
> We often decide on a plan quite early - If we discover it doesn't make sense anymore we should not go on being comfortable not having a plan.

(iii) Using "review" methods with everyone involved was frequently used to make group decisions on projects but can be time consuming and difficult to arrange due to the often complex structure of the team involved.

> We have seen a dramatic drop in defects - 60-70% less just because we do the review.
>
> Good practice is to have at least one weekly UX review for the whole team. These review meetings should be structured (e.g. certain things can / cannot be asked).
>
> Product often thinks that developers should ONLY be involved in last stage. I disagree with that.

Theme category 2: Mixed success in adopting ASD techniques across the organisation

(i) Respondents mentioned that some parts of organisation were regularly using ASD methods with a lot of success. However they frequently mentioned that it not the case everywhere and it was common for teams to fall into a more Waterfall approach in their product development process. It was thought it was due to the different layers of responsibility and decision-making with many stakeholders being involved.

> Working with waterfall - its hard to innovate and make them very efficient and get ideas to the audience. We should make things so they are repeatable.
>
> We (Agile team) work in a very lean way - whereas the rest of the larger organisation (senior management) doesn't want to work in this way!
>
> The more layers of management, the more pressure on top of the persons trickling down information - the more pressure on everyone else.

(ii) One of the challenges of truly adopting ASD methods was the risk-averse nature of the organisation. Respondents talked about the organisation and their teams being nervous to make changes and this would mean that work would take longer to release and thus reduce the success of ASD methods and techniques.

> People think that making something quickly is risky.
>
> But really difficult to change behaviours; get a culture of openness and risk.
>
> Great individuals, I have a lot of respect for them, communicating very well and its just that organisation around them isn't supporting Agile.
>
> It always feels like Editorial is more risk-averse…But I don't think there is anybody in particular – it is us collectively risk-averse.

(iii) The tools and techniques being used to communicate was a common challenge to an Agile way of working. Understanding decisions that had been made and the weekly goals of the team was often difficult when trying to use tools like email and project management tools like JIRA.

> We have JIRA [a project management tool] and classic whiteboard – a Kanban board – it's a problem - you have two sources of truth.
>
> JIRA can be very difficult to visualise.
>
> Sending someone an email to meet is not lean.
>
> A lot of stuff through e-mails that becomes very discombobulated.
>
> Everyone is busy. Lots through e-mail threads. Rather than having quick water-cooler conversations. Especially get more than one other person together. Especially the higher up you go the more meetings. Even if you are lower in rank you tend to have more meetings.

Theme category 3: Desire for more speed and iteration in the ASD process

(i) The stakeholders felt that the organisation was generally too slow at releasing new features to users. Often this was thought to be due to risk-aversion and having complex teams structures, which resulted in more waterfall-like planning and design work happening up-front.

> It should not take us 3 months to appear on the live site. Let's try it and ship it, but we [the organisation] are very nervous of innovating.
>
> We do lot's of big bang releases - but if you try to plan everything up-front you find that it changes very quickly.
>
> Working with waterfall - its hard to innovate make them very efficient and get ideas to the audience. We should make things so they are repeatable.

(ii) There was a desire to try and combat this by using ASD methods such as Kanban to release more iteratively to the user. However, in a lot of cases it had still proved difficult due to the complex organisational structures.

> I am not against designers to have space and time to go away and think. But there is a difference between high-level goals and fluffy designs. Coming up with high-level thinking should be time-boxed.
>
> Pitfall of Scrum – long planning sessions - for developers!
>
> When people see a deadline – they are working towards it! [On a pitfall of Kanban over Scrum].
>
> Lots of micro services can be re-used. Can do more a lot quicker!

5.4.2 Separation between UCD & ASD

It was found from the in-depth interviews that the respondents expressed there was a separation between design and development roles due to the organisational structure that separated the UX design and ASD teams.

Theme category 1: Lack of collaboration due to the organisational separation and structure

(i) The respondents identified that there was a lack of collaboration taking place between designers and developers due to the organisational divisional structure. This was due to the UX design & product teams being structured into separate departments. This makes it challenging for UX designers to integrate themselves in the product team and thus work collaboratively with developers on a regular basis.

> Greatest problem to me: UX is a completely different arm [in the organisation] from product in development. That trundles down the organisation, makes everything very

separate, things like designers being on rotation, breaks the relationship, you have to start all over again.

Ideally we should not come to sprint planning with all the designs upfront. We should be designing / preparing them with the engineers.

The issue is that the designers don't think of themselves as part of the product team.

We don't do enough cross-stream working. Web team doesn't know what's going on in games or apps.

(ii) The organisational separation between UX designers and software developers means that the two roles end up having different pressures and context effects – Designers act as mediators, Developers seen to have to deliver.

Different roles feel different pressures - but it depends on individuals: Don't want to say designers are like this and engineers are like that. Designers are focused on building a good user experience, Developers are more pressured in terms of delivering to deadlines.

If UX are only measured by delivering their piece but not on coding and delivery – then they would get into a silo mentality – because it would be 'the developers fault'.

Product often thinks that developers should ONLY be involved in last stage. I disagree with that.

Theme category 2: Separation between designers and developers causes inefficiency and is perceived to adversely effect quality of output

(i) The organisational separation between UX designers and software developers means that design is worked on upfront in isolation from development and results in heavy handovers of design specifications that developers have not been involved with which causes frustration.

Big up-front designs, locked into pdfs is a waste of money.

UX are not using it at the moment - Developers and UX teams not working as close as they should. Lack of skill sets, bar two designers who are very pro-active.

We do struggle with designs with damaging performance implications - finding that out too late and we always are reactionary. e.g. not functioning components – animation with JavaScript can cause us problems.

I am not against designers to have space and time to go away and think. But there is a difference between high-level goals and fluffy designs. Coming up with high-level thinking should be time-boxed.

(ii) Designers who are working in isolation from developers and handing over "flat PDF mock ups" of designs is a problem because the two roles are not working on the same part a project at the same time. This results in an inefficient workflow between the two roles.

UX waste a lot of their time…Rather than being aligned with the backlog of the Product-owner. Then we would be much more effective as a wider team.

Therefore, the whole workflow of producing something and handing them over doesn't work.

Ideally we should not come to Sprint planning with all the designs upfront. We should be designing/preparing them with the engineers.

We are losing people's ideas, they see it as a huge machine where things flow it, but it seems so long they don't see a point where they can contribute and innovate - Loads of people like small cogs in a machine.

(iii) Due to the lack of integration between UCD & ASD roles in the organisation, there would appear be a lack of togetherness and teamwork between the roles when they did need to come together.

At the moment it seems the other way round – UX identifies some problem – Spends two months designing a solution and then handing it over for a developer to do in a sprint – 'I didn't even ask for this? Where does this come from? Why wasn't I involved in the beginning?'

Failure? If they hate each other.

Doesn't always look like planned. Doesn't deliver value as we thought.

Theme category 3: Relationship is strained with designers involved only at beginning, and developers not invested at delivery

(i) The engineering stakeholders stated that designers would work on solutions without the involvement from their developers at the beginning of the process. They would then handover the solution to developers who would not be ready or not informed about the project.

UX identifies some problem – spends two months designing a solution… And then handing it over for Developer to do in a sprint 'I didn't even ask for this? Where does this come from? Why wasn't I involved in the beginning' - Or – even worse – that UX have interacted with Stakeholders in Editorial before making sure it's gone through the product owner.

The issue is that the UX person treats the delivery team like a customer.

Historically: UX would do all the work -Then its signed off and chucked over the fence to engineering. But when engineering has a problem there is a massive overhead to get an answer on this problem.

(ii) The design stakeholders (creative directors) found frustration when the two roles would come together because designers would find unforeseen challenges when collaborating with developers which would causes changes to be made in the UX that was often rushed or sub-optimal due to a lack of time to follow the UCD process.

What happens in actuality is that the developer will ask the designer 'can you tell me what happens in Sprint 1?'. But invariably what will happen is that the designer says 'This doesn't make sense – we should have done it in another way'.

So a developer is so busy in Sprint 2 that he does not have time to talk to designer. He is basically taken out of the process!

(iii) It was mentioned by the product stakeholders that blame and conflict would occur because of the separation between the two roles at different times during the ASD process.

> If UX are only measured by delivering their piece but not on coding and delivery – then they would get into a silo mentality – because it would be 'the developers fault'.
>
> We are conscious in the past relationship has not been good. We now make sure that at inception of a project developers are involved.
>
> Teams need a common success goal - especially in a small team it would foster people pulling into different directions. It would foster a 'blame' culture!

Theme category 4: Desire for designers to be more embedded in product teams to collaborative with developers

(i) Despite the apparent separation between UCD & ASD in the organisation, the stakeholders all believed that it was possible to achieve better collaboration by embedding designers within the development team.

> Making sure that the whole team is involved. Not UX working ahead, and QA behind. Whole team takes the story all the way through.
>
> We do struggle with designs with damaging performance implications - so that's why it's important to have designers embedded in the development teams.
>
> On communication it comes down to individuals – Some developers are very articulate!

(ii) By having designers more embedded in product teams, it was felt that they could collaborate closely with developers to create prototypes and learn aspects of the development process to inform their design work. The creative directors and technical leads were particular keen on the idea of more prototypes being created as they could be used for design research and usability testing.

> I am mega-keen on pairing UX with Engineers. So we can design & code as we go.
>
> Collaboration works well when one designer can code and prototype and they can build it in code-base. Opportunity for product owner and stakeholders - all about getting feedback as quickly as possible. Unless when a completely new project is starting – then it needs more time.
>
> Designers go on even basic HTML training to help understanding programming and prototyping.

(iii) More integrated designers in the development teams would increase knowledge sharing and mixing of skills. This might include developers getting more involved in the UCD process or designers understanding more about the possibilities and constraints involved in the development of the product.

> Developers understand (now) more about the visual stuff - UX understand that there is a practicality to it! Start design in browsers instead in photoshop where you don't see these limitations until it's too late sometimes.

> Every designer should be able to understand concept of code, HTML, CSS, so that Designers should know the impact of their design. They need to understand the consequences.
>
> It would help a lot if designers would knew these upfront and we would not have gone down this in the first place.
>
> Designers go on even basic HTML training to help understanding programming and prototyping.
>
> Should have a layman's understanding how this thing is coded. UX designer should understand CSS layers etc. They don't have to be developers (because we already have those).

5.4.3 Challenges in decision-making

It was found in the interviews that decision-making in the organisation is a key factor in how UCD integrates with ASD and in particular the perceived effect a "sign-off" culture has on the collaboration between designers and developers.

Theme category 1: Agile collaboration is hampered by the "sign-off" culture affecting decision-making

(i) The stakeholders reported that decision-making was a challenge because of lots of people often being involved. There was frustration of having too many stakeholders influencing the decisions. Despite this not being very true to Agile methods, it was because they felt a responsibility to deliver.

> Many stakeholders are involved: They very quickly spot something they don't like. Even worse - they love the designs – and we cannot change anything. That has happened far too many times.
>
> Stakeholders in Agile trickier (engineering managers, editorial as customer). They want to sign something off. They want to tell us to do something every single day in the week.
>
> The more layers of management, the more pressure on top of the persons trickling down - the more pressure on everyone else.

(ii) Making decisions would often be difficult because different stakeholders might have varying opinions. This need for "sign-off" by stakeholders or multiple members of a team would counter UCD or ASD methods of using data and research to inform decisions.

> Product seems to have final say. Editorial feel they have final say. UXD feel the same. So the levels of bureaucracy – one of the [organisation's] greatest faults.
>
> All these decisions have to involve all these roles, and stakeholders, and the business. So often it seems we are making websites for editorial rather than audience.

(iii) Designers were reported by the technical lead stakeholders to often work alongside other UX teams in order to make decisions. This slows down their work on the product experience and hampers progress with the development team.

> Suddenly UX have to report to Creative Director to get things signed off. They may get pulled off on what the Development team is working on.

(iv) The decision-making process is often slow and involves many layers of hierarchy. This was thought to be because of the risk-aversive nature of the organisation where stakeholders worry about a backlash from users due to the size and scope of the organisation.

> It always feels like editorial is more risk-averse…But I don't think there is anybody in particular – it is us collectively risk-averse.

> It should not take us three months to appear on live site. Let's try it and ship it, but we [the organisation] are very nervous of innovating.

> People think that making something quickly is risky.

> But really difficult to change behaviours; get a culture of openness and risk.

Theme category 2: Collaboration depends on ownership and trust from above

(i) It was believed that the ability to make decisions could be improved by giving more trust to designers and developers by allowing them deliver on their projects, especially in making smaller low risk decisions.

> [The organisation] should relinquish more control. To let people deliver as they want them to.

> We should trust them [designers and developers] rather than forcing them to reporting half the time. We tend to release page types - we tend to think in big pages.

(ii) Stakeholders believed that collaboration could be improved by giving more ownership to designers and developers by letting them adopt suitable and well used processes (e.g., design sprints)

> Autonomy doesn't mean 'doing what you like' - having ownership of HOW you do something - 'flags in the ground'.

> Good practice to enhance design and development is to use 5 day design sprints involving the whole team - this creates ownership from the start.

> Process in design and development more effective and product better if the relationship across disciplines is strong and good.

Theme category 3: Provide "ownership" through goal-setting (e.g. via data analytics and KPIs)

(i) Designers and developers could be given more ownership of their work by being given shared goals to work towards (e.g., KPIs or OKRs). This would allow the

stakeholders to know that they are aiming to achieve what the organisation or product wants but they could make decisions without the need for the same "sign-off" procedures.

Teams need a common success goal - especially in a small team it would foster people pulling into different directions. It would foster a 'blame' culture!

Also the morale of the team is better – more collective investment in it.

What Product should be doing is framing the question - What is the hypothesis we are trying to test here? What does success look like?

There can be individual goals – but they have to lead to one over-arching goal!

(ii) With shared goals amongst the team, designers and developers could decide together how best to tackle the work. It was suggested that this could work by collaborating together to create and test prototypes with the users.

We get some top-level goals but the team can add some specifics: Product owner looks at goals and how to achieve that, Includes UX getting the right designs to help to achieve that.

Designers and developers can prototype something (e.g. click through) and using Userzoom test it on 100 people.

(iii) Shared goals and performance targets like this could provide motivation and empowerment to designers and developers. If they measure their work through data analytics and design research it would give the individuals more of an understanding of the impact of their work.

Let's find out to get value to the audience. Because we really don't know: Does the audience really want XY?

We release it to the audience to LEARN something.

Team are more empowered on data.

(iv) It was perceived that successful work could be defined based on receiving feedback on whether shared goals were achieved. Stakeholders believed that more measurement and feedback like this would allow teams to be more Agile in their design and development process.

Feedback-loop is essential – once you create something - stats & analysis. There some analytical tools like 'Chartbeat'. To look at the effect: "How successful is the thing that went out?

What defines success? New ideas, new things that have measurable value. E.g., sharing content.

Success? KPIs! Success is very measurable…Changes we make should result in improvements. So it's about trying to get the them to solve more problems – that is when they are at their best!

If you can do it (MVT) in one week then that feedback loop is a definite improvement!

5.4.4 Desire for pairing and close collaboration

In the interviews stakeholders talked about the lack of regular close collaboration between designers and developers due to the organisational separation of their divisions but they had experienced isolated cases of proactive designers and developers coming together to collaborate with success.

Theme category 1: ASD is more effective and the product is improved when disciplines are more integrated

(i) It was felt that by stakeholders that designers and developers should work together from the start of a project and be embedded within the same team. This helps with the ASD process as regular iteration and feedback can help to improve the product quickly.

> We now make sure that at inception of a project developers are involved.
>
> We are conscious in the past relationship has not been good. We now make sure that at inception of a project Developers are involved.
>
> Realistically: we should have UX embedded in a team, they work at least a sprint, a scrum in front.
>
> Value we get is when UX are working very very closely with developers in two weeks before Development: When we can feedback very very quickly. With predefined goals.
>
> I am mega-keen on pairing UX with Engineers. So we can design & code as we go.
>
> We do struggle with designs with damaging performance implications - so that's why it's important to have designers embedded in the development teams.

(ii) The organisational separation between the two roles was a problem but it was thought to not be a major blocker in bringing designers and developers together more on projects.

> It's far easier to overcome the difficulties arising from product-based organization. Make sure that people have other ways of interacting with each other in their disciplines.
>
> Should be organised differently. Feels like a different part of the organisation. You have embedded consistent designer who does not get rotated. You have to optimise for product – not the vertical skillset.

(iii) Integration from the beginning of a project and then on-going involvement from everybody in the team helps to provide ownership. An example that was cited by the stakeholders as a useful method for the beginning of projects was to bring people together and use the 'Design Sprint' method as an entire team.

> Good practice to enhance design and development is to use 5 day design sprints involving the whole team - this creates ownership from the start.

Good practice is to have at least one weekly UX review for the whole team. These review meetings should be structured (certain things can / cannot be asked).

Should be organised differently. Feels like a different part of the organisation. You have embedded consistent designer who does not get rotated. You have to optimise for product – not the vertical skillset.

Making sure that the whole team is involved. Not UX working ahead, and QA behind. Whole team takes the story all the way through.

Product often thinks that developers should ONLY be involved in last stage. I disagree with that.

(iv) Stronger integration between the two roles helps to improve their relationships, which improves their work on the product.

The stronger this relationship is, the more effective and efficient is the process/product, and with better results.

We are conscious in the past relationship has not been good. We now make sure that at inception of a project developers are involved.

It's far easier to overcome the difficulties arising from product-based organisation. Make sure that people have other ways of interacting with each other in their disciplines.

UX and development see it more fun and more effective to work more collaboratively. Previously – just making it look like that. Now more collaborative feel – UX frame questions rather than "here is a flat image". Both sides seem like a healthy push-pull like a hand-over.

Theme category 2: Close pairing between designers and developers can improve collaboration and output

(i) It was thought that by pairing together regularly, the output between the designers and developers would improve. In particular, rather than a designer working in isolation to produce mock-ups or wireframes, prototypes and more realistic experiments (e.g., tech spikes) can be created.

Designers and developers can prototype something (e.g. click through) and using Userzoom test it on 100 people.

We had a very good proactive designer, comfortable with prototypes who works with CSS, working with the next one as well and everybody able to comment on it.

Developers should have more time allocated to do some text hikes, prototyping. Getting involved in workshops (ideation, proto-typing) that isn't about delivering something that's going live next week.

Also I pair three of the Devs with some UX and have a rapid prototyping approach.

(ii) By having designers and developers working closely together more iteration could occur in the browser where each role participates in making decisions to fine tune the UX.

It works very well if you have a designer embedded in the team they work a little bit ahead - Close relationship with the developers - Fine-tuning the design and CSS with developers.

Developer & designers 50% of their time on 'lean' very small chunkable things – come up with ideas together. Other future-facing projects – developer should be involved.

Although it's easier for UX to spec something out in separation, in pairs it's easier to see when things look wrong and make decisions on the fly! It makes it easier cutting through the defects.

Developers understand (now) more about the visual stuff - UX understand that there is a practicality to it! Start design in browsers instead in photoshop where you don't see these limitations until it's too late sometimes.

(iii) One of the common problems cited by the stakeholders was the upfront design work without the involvement from developers which is then subject to a handover process that is found to be frustrating. With more close collaboration this handover process is reduced and much more iterative.

UX and Development see it more fun and more effective to work more collaboratively. Previously – just making it look like that. Now more collaborative feel – UX frame questions rather than "here is a flat image". Both sides seem like a healthy push-pull like a hand-over.

It has not worked when we had 6 months upfront design - Completely signed off - and then it took a long time for developers to unpick all the decisions made along the way when all the edge cases have been exposed.

(iv) With clear goals and closer collaboration, designers and developers can iterate together and have ownership and make small decisions together.

Value we get is when UX are working very very closely with Developers two weeks before Development: When we can feedback very very quickly. With predefined goals.

We have UX people right there to explain to Designers what is going on - People have more ownership.

When they feel ownership, empowerment and small enough bits of work to manage them.

Theme category 3: Collaboration and communication from the start increases knowledge sharing between roles

(i) The stakeholders perceived that if the designers and developers spent more time collaborating and getting to know each other upfront, the sharing of their skills and learning from one another would be more frequent and would provide many benefits to the process.

It would help a lot if designers would knew these upfront and we would not have gone down this in the first place.

Technical stack is so complicated now you need to work together - In short time to get a minimal viable product to users as quickly as possible.

(ii) In particular, it was felt that if the roles took time to learn about the basics of the other discipline through training or working closely with the other role it would help collaboration.

> Designers go on even basic HTML training to help understanding programming and prototyping.
>
> Should have a layman's understanding how this thing is coded. UX designer should understand CSS layers etc. They don't have to be developers (because we already have those).
>
> Developers understand (now) more about the visual stuff - UX understand that there is a practicality to it! Start design in browsers instead in photoshop where you don't see these limitations until it's too late sometimes.
>
> Every designer should be able to understand concept of code, HTML, CSS, so that designers should know the impact of their design. They need to understand the consequences.

Theme category 4: Location is an important factor for successful collaboration

(i) The stakeholders had experienced more ad-hoc collaboration between roles when they were sitting alongside each other or in close proximity. It had allowed designers and developers to have discussions and reach decisions together in a more flexible manor than arranging meetings.

> Makes them more pragmatic about each other's areas!
>
> We need these people together at the same time - sit at the same desk.
>
> People can just lean over at a desk and say – 'Can you have a look at this?' If people have to come up two flights of stairs, or have to ring somebody or send an e-mail it makes it more clunky than it needs to be.
>
> Since UX and Development sit together tickets are worked on much more collaboratively.

(ii) It was found with all stakeholders that environment was a challenge for ad-hoc collaboration between roles. Often space was at a premium and there was not a lot of break-out areas where group collaboration and discussions could take place which would make an impact on being able to have regular communication.

> UX team are two floors below - Communication is now harder than it needs to be.
>
> Rooms, office, and locations – working physically - we need more space in general for people to sit. We couldn't fit in another UX person - but if they have space they work well together.
>
> In general space is an issue - we don't have space to put designs up on the walls.

(iii) There was a desire for a environment that was more suited to collaboration where the location of teams could be flexible and suit the needs of a project.

> If teams have more space to put up designs and people passing can just put feedback up as they go past - it would make a huge difference!

> Ideally I would like to sit all my people together - So they could shout at each other, or get up and draw something on the wall together.
>
> It's far easier to overcome the difficulties arising from product-based organization. Make sure that people have other ways of interacting with each other in their disciplines.

5.5 Discussion of Study 1

This study focused on understanding how senior stakeholders (managers) perceive the work of their teams and individuals, how UCD integrates with ASD in their organisation and how UX designers and software developers collaborate and what their pain points are.

From the in-depth interviews a picture emerges of the organisational separation between UX design and ASD teams and the negative impact this is said to have upon the collaborative work between designers, developers and their surrounding teams or processes. The previous section described the findings from the interviews to provide a picture of how UCD integrates with ASD in an organisational setting and how it can impact upon the collaboration between UX designers and software developers. This section discusses these findings to understand how they relate to the literature and the research questions outlined in this study so that the next steps can be formed.

5.5.1 The organisational structure is a challenge for integrating UCD & ASD

The findings show that all of the stakeholders who were in-depth interviewed expressed that due to the organisation's complex structure of divisions, a barrier to successful collaboration had developed because of too much separation between teams and disciplines. In particular the separation was concerned with having UX design teams and ASD teams rather than completely embedded teams. There was a strong desire for designers to be embedded within the ASD teams to remove "silos" and so that regular teamwork and collaboration can occur with developers.

It was also perceived that the organisational culture of "us" and "them" between the UX teams and ASD teams was detrimentally affecting the collaborative practices and relationships between their teams. This meant that without intent, the teams were working on projects at different stages, which would cause frustration and waste time due to a lack of joined-up thinking and knowledge sharing earlier on in the process. This separation and large periods of 'upfront design' on projects would lead to waterfall-like handovers of design

documentation between roles, thus contradicting UCD and ASD principles of facilitating iterative design and development with reduced documentation. This adds to the reported controversy of how much 'upfront design' is appropriate (Adikari et al., 2009). This also fits with previous observations for a need to frequently 're-align' work processes and product development plans (Brown et al., 2011) and that integration of ASD and UX relies on frequent negotiation between these roles (Ferreira et al., 2012).

Due to the divisional structure it was also found that designers were seen to act more like facilitators and developers were seen as the ones responsible for deliver rather than sharing responsibilities, further contradicting the Agile principles around collaborating as a team to get work done to add value to the business and / or users regularly. It is important to discuss that whilst there was frustration from stakeholders about the effects of the organisational separation in the UX and ASD divisions, none of the stakeholders perceived designers and developers to be under performing or not meeting their role responsibilities being asked of them. In their respective teams, the designers were committed to completing work, following their UCD practices and maintaining their contribution to the product or service. Likewise, developers were working according to their own team's Agile processes (e.g., Kanban development) and contributing to their planning, development and evaluation responsibilities.

This was expected behaviour to the stakeholders, as their managers, as there was a lack of organisational-level commitment to guide their collaborations activities with the other role.

The trend was based on the design of the organisation for how it was perceived teams would work best to create suitable user experiences and it was dependant on the stakeholders to make sure that collaboration between teams and roles occurred at the right time during a project. These organisational factors of integrating UCD & ASD aligns closely with the findings from the literature where the integration is shaped and relies on a "complex interplay of organisational and team-level commitments that divide responsibilities between developers and designers" (Ferreira et al., 2012b, p. 18). Despite stakeholders having responsibilities for this interplay between roles, sufficient guidance and understanding for how and when designers should come together was lacking.

Interestingly, when designers or developers did come together in this culture of separation, it would involve them proactively stepping outside of their divisional borders to seek

collaboration. This was known to occur mostly on need-to-know basis by the stakeholders but in some cases closer collaboration had been reported to occur successfully where designers and developers had moved from their predominant team and location to seek each other out or to work alongside each other. This sort of proactive work was often to do with small bits of work where each role actively needed the assistance or knowledge of the other role to help make decisions or contribute towards a piece of work (e.g., a prototype).

Although the integration between UCD and ASD was found to be a challenge within the organisation, when the approaches were followed in isolation they were seen to be beneficial. In particular UCD and Agile methods were viewed with positivity because of encouraging significant flexibility and allowing for projects to be adaptable. However, non-Agile methods were still evident in many teams and roles, with examples of lots of design documentation being used and an "over the wall" culture being cited which aligns with what was found during the preliminary research study detailed in Chapter 2.

Although the organisation structure presents a challenge for integrating UCD and ASD, it is not to suggest that the situation could not be changed and improved. There was a common desire between the stakeholders to embed UX designers within the ASD teams to increase collaboration between roles. It was thought that by doing this more often, knowledge sharing and the mixing of skills would be more common. The creative directors and technical leads of the teams were particularly keen on this as it could help to frequently produce prototypes that could be used for design research and usability testing to better inform decisions. The idea of embedding designers more frequently into ASD teams was encouraged because the stakeholders had witnessed isolated examples of success between the two roles and they believed that personality or discipline differences would not cause any conflicts in practice.

5.5.2 Decision-making is a challenge for collaborative working

The results reflect and add to findings from other studies on factors for successful ASD: the crucial role of decision-making (Drury-Grogan & O'Dwyer, 2013); in the interviews with stakeholders it was felt that the "sign-off" and risk-aversive culture of the teams and organisations was affecting decision-making and the motivation of individuals.

The stakeholders reported this to be because of lots of people often being involved with many layers of decision makers. There was frustration of having too many stakeholders influencing

the decisions and despite this not being very true to Agile methods of giving teams autonomy in decision-making, it was because they felt a responsibility to deliver.
Making decisions was also challenging because different stakeholders might have varying opinions. This need for "sign-off" would often counter UCD or ASD methods of using data and research to inform decisions.

One of the main reasons for decision-making involving many layers of stakeholders and often taking longer than anticipated was thought to be because of the risk-aversive nature of the organisation. As a big organisation with a large audience, the stakeholders stated that people were often averse to being truly Agile or lean in their approach to releasing features and then leaning from their success or failure. Additionally, due to the organisational separation of the UX teams from ASD, designers were reported by the stakeholders to often work alongside other UX teams in order to make decisions. This slowed down their work on the product experience and hampers progress with the development team.

This difficulty to make decision creates a barrier to successful ASD and resides in a crucial component of the Agile philosophy: autonomy and localised decision-making. Other research finds were similar to these findings - Drury-Grogan and O'Dwyer (2013) observed in their qualitative study (focussing on team meetings) that some team members influenced the decision-making due to their seniority or experience. Serrador and Pinto (2015) found that team experience (together with moderators such as quality of vision and complexity of projects) affected outcomes and stakeholder satisfaction.

In contrast to this finding it was clear that stakeholders felt and had experience of collaboration improving in their teams people were given ownership of a project and trust from above. There was a strong desire to provide more ownership in teams and to reduce the risk-aversive culture by using data to inform more decision-making. Respondents suggested that this might include the use of more regular usability testing, statistics, A/B or multi-variant testing methods to give more confidence in decision-making as a team. It was thought that these methods in combination with shared team goals, through the use of Key Performance Indicators (KPIs) or Objectives & Key Results (OKRs) would provide a strong feedback loop and sense of achievement. It was thought that using methods like this might provide a better ability to iterate in a Lean (Lean UX) way by building, measuring and learning as a single focused team or a pair.

5.5.3 Desire for more frequent close collaboration and pairing

The stakeholders that were interviewed claimed that successful close collaboration had been found previously in isolated cases when the following factors were present; Importantly, collaboration depended on ownership of work, trust from above and a relationship between the designer and developer that included sharing knowledge and frequent communication and feedback.

Interestingly, an important factor for successful collaboration was deemed to be that both roles should be involved and working together from the very beginning of a project. Typically, it was common that designers would work in isolation at the beginning of a project without developer involvement and similarly developers did not have enough design input during delivery which was said to frustration due to heavy handovers of spec documents and "flat designs".

It was also perceived by the stakeholders that collaboration between the two roles could be improved through regular pairing with individuals being more proactive to work together, in particular in coming together to iterate together through the development of prototypes that could be used for testing ideas out and helping to make to inform decision-making by the wider team and stakeholders.

Lastly, it was believed that successful collaboration between the roles depended on the use of effective collaboration tools, being in a shared location with each other and having a flexible space and environment to work. These factors correlate with success factors as reported in the literature review and are confirmed here as it was commonplace that these variables were not readily available and evidenced a detrimental effect on collaboration.

5.6 Summary

In summary, this study found that stakeholders (managers) perceived an organisational separation of UCD & ASD teams, highlighting a lack of regular collaboration between UX designers and software developers. This separation between the roles impedes collaboration and causes a problem for efficient working and a quality output. The stakeholders described isolated examples of closer collaboration through pairing and embedded product teams which had provided better integration of UCD & ASD resulting in improved localised decision-

making between roles which generated better solutions through iteration and the creation of prototypes.

The main contributing factors to this included an unsuitable organisational structure which separated the two roles with the use of different team structures, causing too much of a barrier for regular collaboration and knowledge sharing. This structure was also a contributing factor to a "sign-off" culture, thus hampering the ability for the teams to adopt Agile collaboration methods by reducing their ability to make decisions. This lack of localised decision-making had the knock-on affect of the teams and roles being more risk-aversive with their work and suffering from a lack of autonomy of their processes and output.

It was perceived this decision-making problem could be solved through the use of shared goals by using team KPIs or OKRs and data driven work like A/B or MVT testing.

Although ASD practices were seen to be mostly positive, it was recognised that non-Agile, and more 'Waterfall' processes were still common. In particular, the separation between design and development caused inefficiency and large documented handovers rather than regular communication and joined up thinking.

Overall, it was felt by the stakeholders that successful collaboration would be more frequent if UX designers and software developers were embedded in the same team in the same location, if pairing was commonplace and if the two roles shared their knowledge frequently to help make decisions, supported by the use of evidence from data and research. Using the right tools between the roles alongside having an effective and flexible environment for collaboration was also seen to be an important factor for close collaboration.

5.6.1 Next steps

This study gave an insight into the integration challenges of UCD and ASD in a large organisational setting and the effect this has upon collaboration between UX designers and software developers from the perspective of team mangers and stakeholders.

The next phase was to apply a mixed method research approach to understanding directly how the designers and developers perceive the current circumstances – with the aim of

gaining further empirical understanding of the opportunities for collaboration improvements between the two roles.

Chapter 6

Study 2 – Online survey with Designers & Developers

6.1 Introduction

This chapter presents a mixed method study of 109 UX designers and software developers from various teams across a large organisation. Following the results of the 9 stakeholder interviews and with the background research from the literature, a set of questions for designers and developers across the organisation was then derived. Aiming to tap into the commonly experienced issues and the challenges in integrating UCD and ASD methods.

The next section (6.2) presents the background information on the participants, the products they work in and the organisational setting. Next, the data collection and analysis methods are discussed (6.3). The findings from the online survey are then detailed (6.4). These findings are then discussed (6.5) and finally the key findings of the study are then summarised (6.6).

6.2 Background

In this study, designers and developers were asked in an online survey about their roles, perceived level of ASD implementation, satisfaction with ASD, teamwork satisfaction, and perceived quality of communication and collaboration with the other role. In particular, the study aimed to explore whether designers and developers had different perspectives in terms of how ASD works in their organisation, how well the wider team (including other roles, e.g. business analyst) worked together, and whether there were specific issues in designer-developer collaboration and communication that would impact on the current processes, their goals or objectives.

To investigate this an online survey was conducted with 109 respondents from various teams across a large organisation, aiming to capture both qualitative and quantitative insights.

6.2.1 Participants

As discussed in the Research Design Chapter (4.3), the study took place across a large media and broadcasting organisation based in the United Kingdom, with ca. 18,000 employees over all.

To select participants for the questionnaire a sampling method known as 'purposive sampling' was chosen (Palys, 2008). This enabled the focus of study to be on two roles in the organisation, designers and developers. To gain suitable participation and the appropriate amount of quantitative data for this study, designers and developers from across the organisation were asked by email to complete the online questionnaire.

There were 109 participants, (24 women) in the sample. They were asked for a broad job role description to classify them as designers (n=54) or developers (55). They were not asked for their age. The participants were designers and developers from within the organisation. Their work experience varied within their role: For example, some were Junior Designers and some were Senior Designers, yet all were classed as ‘Designers’. All of the product teams that took part in the study have adopted the Agile process to varying degrees of flexibility; some involving mixed approaches to Scrum and Kanban methods.

Of the 109 survey responses submitted, 2 were not analysed as the respondents identified themselves as business analysts or senior management respectively. The remaining respondents were 52 who described themselves as working as designers and 55 working as developers.

Role and Gender	Count	Years Employed	Familiarity
Developers	55	2.23	6.51
Male	48	2.38	6.50
Female	7	1.25	6.57
Designers	52	1.58	5.77
Male	35	1.15	5.71
Female	17	2.44	5.88

Table 6—1: Participant's length of employment (years) and self-rated (1 to 7 scale) familiarity with ASD in each role

6.3 Data Collection & Analysis

Both qualitative and quantitative methods were employed with the goal of gaining a broad insight from across the organisation and its teams and disciplines.

Data was gathered using an online questionnaire tool, called Typeform[6]. This provided user-friendly participation across any device and within a single URL. Questions were answered with both open responses and using the Likert scale method. A a 7-point scale was used, ranging from "Strongly Disagree" to "Strongly Agree" with the middle option of "Neither agree or Disagree".

First, there were basic biographic questions, including on the length of employment. Then, a followed a question on their perceived level of ASD knowledge. There were then 5 items to rate on how well ASD was implemented across the organisation and teams: how well ASD was working in their team, in the whole organisation, in their own product area, how well it served purposes such as achieving successful web based solutions (e.g. responsive web components), and to what degree ASD could be improved.

Then followed 4 questions on teamwork; these questions were taken from Lurie, Schultz, and Lamanna (2011) who validated these. Participants were asked to what degree they were

[6] http://www.typeform.com

encouraged to share ideas, whether they had enough information to do their job well, if the team members make a real effort to understand work-related issues and problems, and to what degree they felt able to act on a team vision. This was followed by 5 questions specifically on the working relationship and collaboration between designers and developers. The questions – derived from interviews with senior stakeholders and the literature review - covered to what degree the two roles worked closely enough together, how productive the working relationships were, if the two disciplines contributed equally and finally if designers and developers had similar skills.

The free-form comments participants gave in in the questionnaire were analysed using a qualitative approach. The comments were imported into a software tool for analysing qualitative data called NVivo[7]. Expressions of opinions, problems, events, reactions and interactions were coded in the text by assigning the piece of text to a category ('node' in NVivo). A category represents a phenomenon, that is, a problem, an issue or an event that is defined as being significant. When categories were found to be conceptually similar in nature they were grouped under more abstract, higher-order categories. Finally, NVivo was then used to create connections between categories and their subcategories. The findings from the qualitative analysis are detailed in the next section (6.4).

For the main quantitative analysis of the survey data, a multivariate approach was selected. This was applied to the rating scores (raw scores ranging from 1 to 7). This covered the 3 main sections of the questionnaire – about satisfaction, teamwork, and collaboration. In this case, the researcher collaborated with an external quantitative research specialist who performed the analysis. The findings from the quantitative analysis are discussed alongside the qualitative findings in section 6.5 of this chapter and the analysis is detailed thoroughly in the published report of this study - see report by Jones, Thoma & Newell (2016) and Appendix B.

[7] www.qsrinternational.com

6.4 Findings

Following the analysis of the qualitative data from the questionnaire, the 'phenomena' were grouped into the following theme categories. These categories are described below with a selection of comments that supports each theme.

6.4.1 Collaboration issues that impede UCD & ASD integration

Theme Category 1: Separation of roles through team setup and location

(i) Both roles perceived a situation of operational separation between the two roles. Where designers are often in different teams to developers - that creates problems in terms of quality and efficiency and often would result in a "them and us" situation.

> UX and development seem very separated
>
> Seems like design and dev are quite silo'd and really don't join up or collaborate together enough.
>
> There is some separation between designers and developers and their workflow.
>
> It currently feels very much like a 'them and us' situation.

(ii) Developers were usually co-located in one designated team, whereas designers were often required to move between different teams and projects. Frustrations were found from both roles about being physically located in separate areas or spaces. This makes collaboration and communication difficult and an agile way of working "almost impossible".

> Collocation and regular collaboration between dev and design (treating eachother as equals) helps a lot. Sadly it doesn't happen often
>
> Until recently UX and dev weren't sat on the same floor of the building we work in, this made working in an agile way almost impossible as we couldn't sit and work together.
>
> Our UX guy needs to be working with us and sits far away meaning constant communication is difficult to do.
>
> Until recently UX and Dev weren't sat on the same floor of the building we work in, this made working in an Agile way almost impossible, as we couldn't sit and work together.

(iii) The environmental setup along with the separation in teams confounded the frustration found amongst the two roles. Where a lack of shared collaboration space causes disconnect between teams and limits any the chance of collaborating in an ad-hoc manor.

> Developers need more whiteboards and space for sketching out diagrams and discussing software architecture. There's not enough space on the walls even for designers to put up

designs. People are squashed up together on small desks and it can be noisy and distracting at times when trying to concentrate.

We don't have a shared work space that works for everyone.

If "shared work spaces" means the physical environment then I think that we could improve some of our office spaces. They're not conducive to visual creativity. For example, it's often not possible to have boards next to desks. Most designers would like to pin work to boards and be able to see it while working.

The workspace in [the building] isn't ideally set up to facilitate this kind of collaborative work. (Impossible to get dedicated space to work together, hot desking, finding walls to put things on...).

Theme Category 2: Too much "upfront" design work

(i) Design work was perceived by both roles to be happening too much in a plan-driven way (i.e. "upfront" before sprints). The consequence of this way of working was that developers felt they cannot engage with the design or contribute early enough.

Designers still working too upfront, not enough communication/feedback.

Agile process is bias towards dev, e.g. UX is an after thought (we're not given enough time to design, or to feedback later, or warning of things coming up). Agile process is focused on delivery, not quality (of UX).

The design process feels waterfall-ish. The designers go away work up some stuff and then bring it back to us to implement.

(ii) Too much upfront design work and the separation of the design and development roles and what they were working on. This results in heavy handovers of design documentation from a designer that isn't inline with the capabilities of development.

Typically, asset handover from designers to developers is a nightmare.

We tend to treat the UX function like waterfall when applying agile to development.

Goalposts or team vision sometimes change quite fast and designs can be left out sometimes. So they put quite a lot of effort to design [almost] pixel perfect designs that will never see the light of day.

The separation in workflow and upfront work means that design work gets handed with little collaboration and often the designs don't line up with what can be built.

Theme Category 3: Not enough joined-up thinking and iteration early on in the process

(i) Both roles believed that their process lacked collaboration early on in the design and development process. In particular, this means that knowledge sharing between the roles is missing and causes problems later.

My own experience from our workflow is that there's not enough collaboration early on between designers and developers. Developers are bit more aware of the capabilities of technology which would be helpful earlier on.

There's definitely a want from multiple disciplines to be more joined up, especially earlier on in the process.

Without collaboration and sharing early on in the process it results in problems later on where decisions have to be changed or more work has to be done.

(ii) The workflow in the current ASD was perceived as not iterative enough. In particular, it was felt (by developers) that the two disciplines were disjointed in their work and processes, which leads to inefficiencies.

The way in which the processes are currently set up doesn't encourage speedy iteration of ideas and concepts.

I think the culture and attitude towards iterative development exists amongst designers and developers, however our ability to work to this is inhibited by product demands, deadlines and multiple layers of bureaucracy.

We're still heavily tied to the business analyst, developer and the test process, things sit in done columns for days before being release to the audience adding zero value while waiting for sign off.

There seems to be a significant disjoint between the functions required for a leaner approach to UX required in the Agile environment.

(iii) Developers reported that they would like to be more involved in the ideation phase and involved earlier in the process.

Would be helpful to see designs (even just wireframes) earlier in the process.

Frequently devs are not able to take part in the ideation process because they are assigned to other work - the Product team require them to continuously write code and build stuff rather than getting involved in the early design phase.

Historically developers haven't had a chance to participate in the ‘design’ process.

Theme Category 4: Perception of a lack of regular communication between the roles

(i) Respondents also complained about a lack of regular communication and collaboration between designers and developers. This meant that teams felt software development and UX work was often not aligned.

Our UX guy needs to be working with us and sits far away meaning constant communication is difficult to do.

The development team, with product and project works in a very Agile way. However, we often find when it comes time to start development on a story that the UX assets we need are not ready.

I strongly believe that communication and relationships between developers and designers is crucial. Unfortunately, that's not happening at the moment.

Until recently UX and Dev weren't sat on the same floor of the building we work in, this made working in an Agile way almost impossible, as we couldn't sit and work together.

(ii) Both roles felt that because of a lack of communication between the teams, it mean causes them to often be out of sync with each other and what they were working on.

There is very little 'dialogue' involved in agile between us so we don't work on the same thing together at the same time, or least we aren't across what each person is doing.

We're still kept a bit in the dark about what's going on.

(iii) Designers felt more out of the loop than developers in terms of wider team information, most likely because they need to work across (or change) teams more often.

Agile process is bias towards dev, e.g. UX is an after thought (we're not given enough time to design, or to feedback later, or warning of things coming up). Agile process is focused on delivery, not quality (of UX)

The product and developments teams appear to have all the information they need at their disposal, but the design teams have little influence over this or how it works.

It would be useful to link UX to the product roadmaps earlier so we know what is going on but this doesn't happen and we have to work across lots of teams.

Historically the focus on new features has been developer centric. Agile practises starting after the design has been completed and the "build" begins.

(iv) It was perceived that communication tools were being used ineffectively for cross-location working between designers and developers, resulting in inefficiencies.

Lots of time is spent replying to issues in JIRA or on slack when it would be much quicker to have a conversation in person.

I just feel there are too many tools and communication tools.

Slack has been a great tool for communication between designers and developers at different locations but communication isn't great and can be much improved.

6.4.2 Perceptions of other roles working style and the ASD process

Theme Category 1: Difference in understanding of each others roles

(i) Developers perceived that designers do not know enough about the technical limitations when designing solutions. What this meant was that developers perceived inefficient and redundant work efforts at times during the ASD process.

It would be useful to engage with developers earlier in the design process to gauge what is possible and come to a consensus on a practical way to iterate.

A major issue is designers not knowing what can and can't be done, and the implications of certain design decisions.

Working together so designers understand the key technical challenges of developers is also very important, a tiny tweak in designs can shaves days (or even longer) in development costs.

(ii) Designers felt that they were not exposed to enough information or the UX work was not central enough in the development process. Overall they perceived that the quality of the teamwork was an important part in the success UCD and ASD process.

Historically the focus on new features has been developer centric. Agile practises starting after the design has been completed and the "build" begins.

This means that the overall direction of our designs is constantly driven by external requirements. There is very little 'dialogue' involved in agile design.

Agile process is bias towards dev, e.g. UX is an after thought (we're not given enough time to design, or to feedback later, or warning of things coming up). Agile process is focused on delivery, not quality (of UX)

The relationship between design and development could be improved by involving UX into the Agile process and making the Agile process more designer friendly.

Theme Category 2: Developers frustration with lack of access to designers and too much upfront 'design vision'

(i) Developers perceived designers to be focused too much on the 'design vision' (often portrayed as "flat mock-ups"). These are usually static webpage prototypes with worked-out graphic elements (fonts, headers, etc.) and are therefore considered to lack necessary details about the interaction layer in-browser.

The design teams here always seem to spend a long time designing for the 'Full Fat' version of the product - the ideal version which has all the features and the best user experience.

There still seems to be a resistance from many designers to move away from drawing pictures of websites in Illustrator and get things in the browser.

Often UX do big up-front glossy designs. Sometimes the designs aren't complete.

(ii) Developers also felt that there was a lack of access to designers during the ASD process between the two roles and they questioned the contributing skillset of designers. In particular, there was an interest in the how the ability to participate in aspects of coding could potentially assist with different parts of a project.

Don't think many designers code. Could be handy in some situations.

The coding skills of a designer are different, a designer will be more focus on the front-end development but not necessary how well the code is written, our main objective is to make things visually consistent and interactive.

Some of the designers that I heard of can code too which could be really helpful on a lot of projects as we could work more closely together.

(iii) Despite developers interest in the skillset of designers and how the roles could be more joined up during the ASD process, they was a belief that both roles were important disciplines and contributed equally to the projects. Importantly, their emphasis was on how the roles could be more aligned in terms of collaboration.

I think designers and developers contribute equally but differently to projects.

In general, I don't think the aim should be to push designers to become coders nor developers to become designers. These are two separate disciplines that require time to keep on maintaining, improve in etc… I think there should be a closer collaboration between both, designers and developers, so that both reach their optimum potential within their disciplines. I think this is what will yield amazing user experiences that will grow on users and make an impact in the digital world.

We can be more collaborative and combine our skills as we share more knowledge more often.

Theme Category 3: Differences in perception of satisfaction in ASD process and implementation

(i) Overall, both roles were aligned on feeling positively about the ASD process. However, both were critical of its current implementation in their teams and across the organisation. Overall they perceived that their processes could be improved or better aligned to suit the organisational requirements.

Agile is great. I'm in to it. But its not the only tool in the box - and often I feel like it's being used for everything, which gives us a 'screw and hammer' situation.

Agile is a term widely used but not properly implemented within the [the organisation] due to its scale. Although I think Agile is the right approach to working with....I don't feel like these things are working here.

The relationship between design and development could be improved by involving UX into the Agile process and making the Agile process more designer friendly.

Generally the design and development process could do with some improvements.

Also some of the problems that exist are not all the fault of designers and developers. A root cause of a lot of these problems span from product level and stakeholders who still expect to follow a more waterfall process (whilst still calling it Agile).

(ii) Overall, both roles perceived the ASD process to be at odds with UCD and often out of sync with the process commonly found to be more developer-centric than user centred. This adversely affected the UCD process and was acknowledged to be a problem by both roles.

Agile process is bias towards dev, e.g. UX is an after thought (we're not given enough time to design, or to feedback later, or warning of things coming up). Agile process is focused on delivery, not quality (of UX)

The development team, with product and project works in a very agile way. However, we often find when it comes time to start development on a story that the UX assets we need are not ready.

Historically the focus on new features has been developer centric. Agile practises starting after the design has been completed and the "build" begins.

(iii) Designers perceived that ASD satisfaction came from feeling part of the wider team throughout the design and development process and in particular being involved in the decision-making dialogue with team members. Without this involvement, often design decisions were changed or scrutinised by the wider team due to a lack of involvement.

This means that the overall direction of our designs is constantly driven by external requirements. There is very little 'dialogue' involved in agile design.

The Design side of the output is open to scrutiny form everyone as everyone has an opinion as they have eyes. The Dev side isn't scrutinised at such a microscopic level.

Its good to be involved in the full ASD team throughout the process but this doesn't always happen (or its not in our control) and results in a lack of collaboration and our decisions get questioned.

Whilst I pair with developers and go to their stand-ups on a daily basis, it still feels like we (designers) work in more of a waterfall manner. How can we integrate more fully with their process and vice versa? - It currently feels very much like a 'them and us' situation.

6.4.3 Desire for closer collaboration and sharing between roles

Theme Category 1: Need for more knowledge sharing and combining of skills

(i) There was a strong desire from both roles to have a greater overlap of skillsets and less division or "silos" between designers and developers. In particular it was felt that if the two roles worked more closely together, more knowledge sharing would be beneficial in achieving shared goals.

I think there's a clear division between the skills of designers and developers. Some designers can code, but only a few. I think that is fine. We should combine our skills working together to achieve the same goals.

I feel there is often the natural tendency to silo certain skills, developers code, designers wireframe. Although I feel a blurring of the borders would be beneficial.

We need more pair design/programming. Red lining is time intensive and ineffective! (i.e. takes forever and the devs don't match it closely anyway). Front-enders should be more involved in design and designer more involved in the dev.

(ii) Both roles want more frequent sharing and combining of their skills. In parricular, where more prototyping can occur between the roles as a shared partnership. It was perceived that this would help to create a stronger team ethic and greater knowledge of the different disciplines.

> While occasionally some designers are working with code, it would be nice to have more of an overlap with developers through a larger amount of HTML prototyping. It would also be useful if these prototypes could be constructed in a way that could be taken and built upon by developers rather than having to start fresh.
>
> When this works it's a partnership that's formed, both sides have a mutual respect and understanding of each others disciplines, there's a decent working knowledge of each others skills.
>
> Would like to be able to sit with a developer and prototype something, but they're all tied up working on tickets to build red lines designs, instead of working alongside you on getting an easy to build, well UX'd product designed in the first place.
>
> I've had the opportunity to work with some young developers who are really good at their craft and keen to work closely with designers. This helps push the product forward and create a strong team ethic.

Theme Category 1: Desire for closer proximity and to be part of the same team

(i) Close proximity in terms of location with the other role was seen as particularly important to enable side-by-side communication and regular ad-hoc discussions. This would enable more ideation and problem solving through sketching and discussions together.

> Nothing quite beats sitting with them [designers] and talking face-to-face and sketching ideas and thoughts together.
>
> It is also important to have design sitting closely to Development and product so that ideas can bounce faster between each other's teams.
>
> I think the more designers and developers work closely together, the better. If designers were co-located with project teams, that would be incredibly useful.

(ii) Not only do designers and developers want to work in close proximity with each other but there is also a desire to be part of the same embedded team together.

> Design always feels separate. It works much better when designers are embedded within the same team as developers.
>
> Our team are lucky enough to have UX designers embedded with the team which is extremely helpful for iterating on UX changes.

Theme Category 2: Iterative development through close collaboration and pairing

(i) The designers and developers perceived that pairing together on projects was beneficial to their design and development processes. In particular, by having a close working partnership it enables effective and iterative working on the product.

> For me, the only way to successfully build great responsive products in an iterative way is for the developer and designer to collaborate on the product itself (i.e. the production web site). This means sitting next to each other, and focusing most of their time on the product itself and less of their time on static mock-ups / prototypes.
>
> Close working, and effectively pairing with designers and developers is what I think works best.
>
> I think the more designers and developers work closely together, the better
>
> Working in a Design & Dev pair on every new feature from the beginning works great.

(ii) By working closely together as a pair, there were many benefits that were expressed by both roles. Not only does it allow the work to be more iterative but sharing knowledge is more frequent and a better understanding of the requirements can be found.

> Working together so designers understand the key technical challenges of developers is also very important, a tiny tweak in designs can shaves days (or even longer) in development costs.
>
> I think designers and devs should still work closer on doing rapid development after initial designs have been done. This would mean faster feedback from each other and from the audience.

Theme Category 4: Localised decision-making and joined-up thinking earlier on in the process

(i) Developers were usually co-located in one designated team, whereas designers were often required to move between different teams and projects. This setup meant that planning, design and implementation requirements were not always easy to coordinate between the two roles. There was consequently a strong desire for more "joined-up" thinking and discussions between the two roles to understand the constraints and to "gauge what is possible".

> More workshops or fun things at the beginning of the project or feature to encourage designers and developers to work together
>
> I would suggest that the way we are currently working in sport is good example of how design & development process should work. Working in a Design(UX) & Dev pair on every new feature from the beginning works great.
>
> It would be useful to engage with developers earlier in the design process to gauge what is possible and come to a consensus on a practical way to iterate.

> For me, the only way to successfully build great responsive products in an iterative way is for the developer and designer to collaborate on the product itself (i.e. the production web site). This means sitting next to each other and focusing most of their time on the product itself and less of their time on static mock ups/prototypes.
>
> I have worked on one project when the dev was involved in the design phase and this was a great success as he contributed some really useful ideas and was able to create prototypes using real data.

(i) Finally, there was frustration about the "sign-off" culture and decision-making in the organisation. Due to the structure of the teams, there is a clear hierarchy of product owners who make decisions, and often even these have to be deferred to higher levels (e.g., creative directors). This pre-ASD legacy was perceived as preventing the teams from being autonomous.

> There is also a major delay in decision-making - as in, this holds up projects because people seem reluctant to make decisions on things.
>
> I feel there are too many stakeholders and managers involved to work in an Agile way. Getting signoff when there are so many people involved and juggling so many conflicting interests and ideas can be time consuming.

Overall, the quantitative analysis of the ratings scores (Jones et al., 2016) reflects the qualitative analysis findings described above. In particular, the quantitative analysis found out as to which degree variables such as length of employment, agile knowledge, teamwork satisfaction, quality of work environment, and quality of teamwork would correlate with (and therefore predict) overall satisfaction with the agile development process.

The two groups did not differ in their perception to what degree ASD is already applied in the development process, both on team level and in the organisation as a whole, and with what success. Both also roles thought and agreed that there was significant scope for improvements of the ASD process and overall satisfaction scores trailed scores for ASD satisfaction within people's own teams.

In the analysis of the quality of ASD setup in the work environment - there was a significant difference between roles in the perception of how well the physical environment is set up to support collaboration in ASD. For developers the main factors for a successful ASD process implementation were environmental setup and collaboration with designers, for designers the main predictor was the perceived quality of the (wider) teamwork. These findings are interesting because of the lack of overall differences between designers and developers in their assessment and perception of ASD processes in the organisation.

6.5 Discussion of Study 2

The current study focused on finding out about the collaborative relationships between designers and developers in a large media organisation working within an Agile software development environment.

Overall, the findings in the survey confirm some of the previously reported factors of successful ASD implementation, but additionally show which of these aspects vary across the two roles. There were differences in how designers and developers perceived successful teamwork and collaboration. This is despite the fact that designers and developers worked in the same organisation (and overall location) and although they appeared to be generally aligned on many questions around agile processes and its current implementation.

For both roles, the findings show that the surrounding organisational structure hinders their ability to work closely together on a regular basis. This was thought to be because of the separation in their teams where they were placed in structurally different divisions within the organisation. This separation, both in terms of physical location and via "silos" in team structures was felt to lead to a lack of regular overlaps where knowledge and practices can be shared and joined-up, especially early on in the process. This reduced the crossover of skills and awareness of their work and correlates with Sy and Miller (2008), who describe that non co-location of the roles results in collaboration barriers and communication difficulties.

Furthermore, the separation and a lack of close collaboration between the two roles in the organisation are demonstrated with frustrations about disjointed processes and too much "upfront" design work. In particular, it was felt that design work was perceived by both roles to be happening too much in a plan-driven way where the work is handed over to the developers via "red-lines", "flat-mockups" or "mark-up" files. This was found to be inefficient and time consuming for the developer to retrospectively understand the design details with an absence of the designer from their day-to-day work and caused frustration. This contradicts with the UCD and ASD principles of facilitating iterative design and development with reduced documentation and adds to the reported controversy of how much 'upfront design' is appropriate (Adikari et al., 2009). This also fits with previous observations for a need to frequently 're-align' work processes and product development plans (Brown et

al., 2011) and that integration of ASD and UX relies on frequent negotiation between these roles (Ferreira et al., 2012).

From the respondents' comments it seems clear that a solution to these problems resides in improved communication and collaboration between the two roles. The analysis suggests that agile satisfaction for both groups is associated with the frequency of working in pairs (and increased physical co-location), adoption of a more iterative workflow (including design iterations), and furthermore a more localised (less hierarchical) decision-making process. This also fits with previous observations for a need to frequently 're-align' work processes and product development plans (Brown et al., 2011) and that integration of ASD and UX relies on frequent negotiation between these roles (Ferreira et al., 2012). These solutions are in contrast to the delays and organisational separation being experienced because of design decisions being "signed-off" by senior stakeholders, a tradition inherited from pre-Agile structures. This aspect of the current process is of course almost a direct contradiction to purist conceptions of ASD, which demands self-organising teams that can take decisions and drive development largely autonomously.

Overall, the quantitative analysis confirms and reflects the qualitative findings and in particular showed significant differences in the each roles satisfaction with ASD and its implementation in the organisation (Jones et al., 2016). Developers' satisfaction with ASD correlated with access to and collaboration with designers, as well as the environmental (physical) setup at work. Designers' satisfaction with ASD, however, was associated with the perceived quality of teamwork. This distinction was also reflected in the overall differences – whereas designers were less satisfied with overall teamwork, developers scored lower on satisfaction with designers. Across both sets of analysis, a particular difference between designers and developers was their perception of having enough information for successful teamwork. Designers rated this lower, possibly because they were often not constantly embedded in a team, or not co-located with their team members. Interestingly, the widely reported important factor of differences in experience with ASD (e.g., Drury- Grogan & O'Dwyer, 2013; Serrador & Pinto, 2015; see also Vijayasarathy & Turk 2012), does not seem to have any impact on our measures here: both the length of work as well the explicitly elicited (self-assessed) knowledge of ASD were found by the quantitative analysis to have no significant effect on moderating satisfaction with the development process.

The observation that the analysis did not find an effect of length of employment or ASD experience – at least on the quantitative data – is important. It may be due to the fact that in the sample of designers and developers were often already working in agile teams (and possibly due to the training provided in the organisation), therefore on average their exposure or knowledge was possibly already close to ceiling. In other words, the targeted sample of already rather dedicated agile operators may have highlighted more persistent issues that are to do with often-intractable barriers to ASD inherent in organisational structures and culture.

Other research finds – similar to the observations – that barriers to successful ASD reside in a crucial component of the Agile philosophy: autonomy and localised decision-making. Drury-Grogan and O'Dwyer observed in their qualitative study (focussing on team meetings) that some team members influenced the decision-making due to their seniority or experience. Serrador and Pinto found that team experience (together with moderators such as quality of vision and complexity of projects) affected outcomes and stakeholder satisfaction. Unlike other research (see a review by (Jurca, Hellmann, & Maurer, 2014a) the teams did not perceive UX work as optional, which may be due to organisational emphasis on UX work and general changes of attitudes in the industry over time.

Another barrier in the working relationship between designers and developers – which is have not addressed here - may also lie in their different personalities. There are reports of differences in personality and style within software development teams (Capretz & Ahmed, 2010). Acuna, Gomez, & Juristo (2009) found that when student teams adopted Extreme Programming (XP) they decided on their own type of cooperation and they experienced the least conflicts and showed higher levels of job satisfaction. However, it is not clear whether software engineers are different from other groups. Beecham (Beecham et al., 2008) found in a review of 92 papers that just half of studies report that engineers are distinguishable from other occupational roles in terms of motivation. Vijayasarathy and Turk (2012) emphasise the importance of 'enabling factors' such as training and setting norms in the Agile environment are important for its success.

6.6 Summary

In summary, Study 2 shows that the relationship between UX developers and software designers is an important factor for successful UCD work in an ASD process. However, the determinants and long-term barriers for successful ASD in this relationship are still undetermined and should be explored further alongside new collaboration methods and practices.

In this study designers and developers were asked in an online survey about their roles, perceived level of ASD implementation, satisfaction with ASD, teamwork satisfaction, and perceived quality of communication and collaboration with the other role. Developers rated the main factors for successful ASD work as having access to designers and an optimal environmental setup. Designers' main concern was related to improving wider teamwork, in particular the sharing of information. Both groups indicated that close collaboration and informal communication between designers and developers was desired. Respondents also perceived senior stakeholders' hierarchical decision-making process as a barrier to successful ASD. Co-location and pairing of designers and developers was therefore also seen as an opportunity to enhance more localised and autonomous decision-making.

6.6.1 Next steps

Along with the findings from stakeholders in Study 1, this study found that both roles perceived collaboration could be improved. In particular, co-location and pairing of designers and developers was desired as an opportunity to enhance more localised and autonomous decision-making, especially early on in projects.

Motivated by this finding, the next study further explores gains and challenges of collaborative pairing between the two roles in UCD and ASD processes.

Chapter 7

Study 3 – Pairing Study A

7.1 Introduction

In Study 1 & 2, both roles and their stakeholders perceived that collaboration could be improved. In particular, co-location and pairing of designers and developers was desired as an opportunity to enhance more localised and autonomous decision-making, especially early on in projects.

Motivated by this finding, Study 3 further explored the gains and challenges of collaborative pairing between the two roles adopting UCD and ASD processes in a large organisation. This chapter presents an ethnographically informed study of a UX designer and software developer in a large organisation that worked as a pair together for a period of 1 week. A questionnaire at the beginning and at the end of the study period captured the insights about their experiences.

The next section (5.2) presents the background information on the participants, the team they work in and the organisational setting. Next, the data collection and analysis methods are discussed (5.3). The main themes that emerged from the questionnaire are then detailed: These findings are then discussed (5.5) and finally the key findings of the study are then summarised (5.6).

7.2 Background

Alongside the preliminary practice-led research and the review of the literature, Study 1 & 2 found that the relationship between developers and designers is perceived as an important factor for successful UCD work in an ASD process. However, the determinants and long-term barriers for successful ASD in this relationship in an organisational setting are still undetermined and should be explored further alongside new collaboration methods and practices.

Respondents in Study 1 described the integration between UCD and ASD as a challenge because of the organisational setup, where the UX team(s) predominately were separated from the Agile product teams. The stakeholders in Study 1 along with the respondents in Study 2 also perceived that senior stakeholders' hierarchical and risk-aversive decision-making process as a barrier to successful ASD. Co-location and pairing of designers and developers was therefore also seen as an opportunity to enhance more localised and autonomous decision-making by both the stakeholders and the designers and developers themselves. Respondents in Study 1 & 2 indicated that close collaboration and informal communication between designers and developers was desired.

So in Study 3, to further understand the determinants for close collaboration and pairing in an organisational setting a UX designer and a software developer were paired together within their normal environment. Like the aforementioned research, the setting was part of a large Agile organisation within a product design and development environment that typically followed an ASD process. The office was an open plan space with banks of desks, break-out-areas for ad-hoc meetings and collaboration and meeting rooms that are available to book out.

This study took place over the course of one week (five working days) and was described by the team as an "Innovation period". This meant their normal team structure and ASD process that was typically in place was non-existent. Instead, the designer and developer had freedom to work together as a pair, thus providing an opportunity to learn about their approach and their collaborative practices. The aim was to capture insight and further validate the previous research about the perceived problems with the current integration of UCD and ASD in an organisational setting. In particular, the aim of this study was to understand how they currently collaborated together in their team(s) and how UCD integrates with ASD and how their practices change during the study by following a paired approach.

7.2.1 Participants

To select participants for the study a sampling method known as 'purposive sampling' (Palys, 2008) was used. By using the directories of employees within the organisation designers and developers could be selected by the researcher and contacted for participation. Two participants were then selected.

Both of the participants were part of the same ASD product team in the organisation and were at the same job role level as a User Experience Designer and Web Developer respectively. They had been working in the wider product team for a period of 1 year and had worked together as part of the ASD team which developed web based software using the Agile Scrum methodology. The participants experience in their roles varied but they had both worked at the organisation for a similar period of around 2.5 years. The designer was responsible within the team for following a UCD process alongside other colleagues to design the UX for the website. The developer was responsible for implementing features onto the website as working software.

7.2.2 Projects

For this study, both the UX designer and the software developer were focused together on the same project which was to design and implement a new type of music index for the web-based product. This took place over the course of one week and where they could spend focused time with each other. This approach contrasted with their normal project work. Typically, the UX designer would be working across multiple projects in the product and the developer would be focused on implemented a specific set of features.

7.3 Data Collection & Analysis

For this study, qualitative methods were employed to gain an insight and understanding about the pair and their work. To capture an understanding and insight into the successes, failures and outcomes of the study a questionnaire was used. The study lasted for one week, and the pair was given set of questions within a questionnaire that they were asked to answer before and after the 'innovation period'. The aim of this was to gain an honest of interpretation of their perceptions and views.

In the questionnaire, the first set of questions began with a basic set of biographic questions, including the length of their employment and how long they have been in their role. Then followed three open ended questions to gain an understanding of any current perceived problems of how they currently work, how they will approach working with the other discipline and how working together could improve their efficiency and output for the audience. Following the week together the pair were presented with another questionnaire to

capture insights about how they felt their work together went. Three questions were asked to understand the successes of the pairing and if any problems were found, how the work and their practices and processes modified or changed if at all and finally a question to find out if their efficiency and output to the audience changed or improved.

The participants were also encouraged to note down and record observations, problems, reactions or circumstances that was relevant to the study during the week to help them to answer the questionnaire at the end. Its important to note the participants were asked to not discuss their answers to the questions prior to responding to the questionnaire. The questions asked of the participants are listed below in Figure 7-1.

Pairing Study A - Questionnaire

Before the pairing study

- From your perspective, what are the current challenges surrounding the process and practices between UX designers and software developers?
- How do you think you will approach working alongside the other discipline? i.e. Provide an overview of your proposed workflow.
- How do you think working together will improve your efficiency and the output for the audience / user?

After the pairing study

- How did it go? Did you come across any problems and what were the successes?
- How did you change your own workflow and how did you influence the practices / process of others?
- How did working together improve your efficiency and the output to the audience?

Figure 7-1: Pairing Study A – Before and After Questions

7.3.1 Qualitative Analysis

This analysis is separated into two sections, before and after the pairing study. This will provide a comparison and understanding about the study outcomes and developments in practice and behaviour between the pair. All of the questionnaire responses participants gave in Typeform were analysed using a qualitative approach. The comments were imported into a software tool for analysing qualitative data called NVivo[8].

Expressions of opinions, problems, events, reactions and interactions in the text were coded by assigning the piece of text to a category (''node'' in NVivo). A category represents a phenomenon, that is, a problem, an issue or an event that is defined as being significant. When categories were found to be conceptually similar in nature they were grouped under more abstract, higher-order categories. Finally, NVivo was used to create connections between categories and their subcategories, which are listed below.

7.4 Findings

7.4.1 Team collaboration & integration issues

Theme category 1: The organisational separation of the two disciplines hinders collaboration and knowledge sharing

(i) Prior to the pairing study both roles perceived that the organisational structure made collaborating a challenge between the two disciplines. This was thought to be because of the separation in their teams in that they existed as separate discipline-based divisions within the organisation.

> The problems are not really within the two disciplines. The problem is that they are separated as two disciplines.

> I think we should all work in each other's worlds all the time and we would become more conscious of the overall process and start to think in more, sophisticated, compatible ways.

[8] www.qsrinternational.com

(ii) The cause of this separation was felt to lead to a lack of overlaps where knowledge and practices can be shared and joined-up. This would reduce the mixing of skills and awareness of each other's processes.

> Designers are unaccustomed to thinking about how things are constructed, not because they are incapable of it, but they are not asked to think about it so that part of their thinking goes untrained.

> Developers do not work in a design environment so they are unaccustomed to being presented with open briefs, they expect more definition. This is not because they cannot cope with it, just that they are not used to it.

Theme category 2: The "handovers" of design documentation and specifications impedes efficiency and collaboration

(i) Design work was perceived by both roles to be happening too much in a plan-driven way where the worked is handed over to the developers via specifications or "mark-up" files. This was found to be inefficient and time consuming for the developer to retrospectively understand the design details with an absence of the designer from their day-to-day work.

> The most precise method of delivering designs - full mark-ups - is incredibly time consuming, requires handholding post-build when developers inevitably miss the details (there are hundreds of them), and also have the unfortunate effect of acting like a contract - that is, if the designer makes a mistake or an omission, it's difficult to correct without it becoming another 'backlog item' that is forgotten.

> The current process still feels too much like a handoff rather than collaboration. More often these days the developer gets involved with shaping and prototyping in the UX space, but it still feels like we get handed a PSD, some vague notes and that's about it.

Theme category 3: A dissatisfaction about the tools used to complete work

(i) The designer voiced issues that arose from the use of tooling that hampered the completion of work and collaboration between the two disciplines. In particular, it was felt that it is difficult without the specific knowledge or skills to become integrated with development processes. Despite this, it was acknowledge that neither the design tools nor the development roles being used were useful for both roles.

> Tools and processes that are used by developers are not 'designer-friendly' by their nature. Sandboxes, dev certs, version control, stacks and frameworks... All barriers to a designer (despite one that's fairly code-savvy) in contributing to a project without intricate knowledge of the processes involved.

> The web is our medium, and we're not using tools fit for purpose. We don't make it easy for developers in terms of communicating our designs (things get lost in translation from static mock-ups to code, and developers get the blame when things don't turn out they way they were intended).

7.4.2 Desire for closer collaboration and faster iteration

Theme category 1: Workflow in the current process could be more iterative and connected

(i) There was a desire between both roles for the design and development process to be more aligned to ASD principles by being more iterative in the development process.

In particular the pair were frustrated by the speed at which their releases occurred due to their separated workflow and they believed that by working side by side they would get releases out to their users quicker because it could be more throwaway and flexible.

> Building is thinking. If it doesn't work, trash it. Do another… repeat, repeat. Try to apply the diverge-converge process to building. Code should not be treated as precious; it's cheaper than paper so rip it up and chuck away. repeat, repeat.
>
> The process of reacting to audience / editorial demand and creating new things and getting them out would be quicker and more flexible.
>
> We will truly be iterating because we're designing and building in parallel.
>
> The iteration is much tighter and faster, yielding a better solution the first time through.

Theme Category 2: A desire for more prototyping to improve handovers and create "back-and-forth" collaboration

(i) Prior to the pairing study, there was a strong desire between both roles for prototyping to be a more commonplace within the ASD process, especially early on in a project. This was due to the problems and frustrations of the current handover process between designers and developers that involved design mock ups and specifications. The pair believed that spending more time building prototypes would be a better investment of their time as it could be tested with users or help with speeding up the overall build process.

> We should invest more time in prototyping in the creative process, and less in pixel-perfect photoshop mocks.
>
> I would also expect us to avoid creating 50-page documents full of marked-up styling for this project, and instead see code-based prototypes as our design deliverables.
>
> There's something motivational about bouncing prototypes back and forth, and I've already seen this impacting the project in its early stages. Developers are encouraged to be a part of the design process from the beginning, and through prototyping/hack days and general back-and-forth collaboration, there's more a sense of a unified team than I've experienced elsewhere.

> We are putting emphasis on prototypes as a catalyst between design and dev. An example of this would be a code-based design prototype that is passed on to a developer to borrow the CSS and build 'for real' on test environments.

The analysis of the comments before the pairing study reflects some of the difficulties found between the two roles in Studies 1 & 2. This is shown through the organisational separation between UX & ASD teams and frustration about the handover process and the mixed use of tools and practices resulting in a lack of successful collaboration and a workflow that is slowed and is perceived as being inefficient because of these issues. There is also a desire for prototypes to be more commonplace as an output between the two roles to act as a talking point rather than a "flat mock-up" of the experience.

The problems with the handover process between a designer and developer have been highlighted multiple times and it appears to be conscious issue between these two participants with the separation between the two highlighted by a lack of joined-up thinking and knowledge sharing. With this in the mind of the participants there is a clear desire between them to proactively improve on these issues going into the pairing study, with both participants wanting to improve their process and workflow with more prototyping and a more iterative, faster output.

As described in the Data Collection & Analysis section (7.3), the pairing study lasted for one week, after which the designer and developer responded to the questionnaire. The previous qualitative analysis method was then applied to the questionnaire responses and the findings are detailed below.

7.4.3 Improvements through close collaboration and pairing

Theme Category 1: A higher degree of knowledge sharing and joined-up thinking

(i) The pair mentioned that because of pairing and increased side-by-side communication, the sharing of their knowledge and skills was increased during the time they spent together. This collective sharing was valued because it enabled the mixing of their skills to improve their work. One example was that the designer found a greater understanding of performance of the site and the design implications upon the user experience were enhanced with this in mind.

> I think we learnt from each other - I certainly learnt from him!

> Working closely together allowed us to quickly identify each others' strengths.

It forced me (designer) to think seriously about performance in transitions etc, something that usually we would palm off to a developer.

(ii) Knowledge sharing and overlaps between the pair was improved by being together on their project throughout and in particular from the start. Allowing them both to understand each other's skills and benefit from working side-by-side.

It did not feel like a design/dev collaboration, it just felt like how work should be. Making things in a group of people who have a range of skills.

Working closely together allowed us to quickly identify each others' strengths. We shared making the front end code (at least CSS), though that may not have been entirely planned beforehand.

Going back to a paper-based board of work to do actually felt really refreshing and satisfying when completing items. I really got a lot out of hearing the reasons behind decisions in layout and style, and also contributing to those discussions together side by side.

(iii) In additional to increasing their knowledge sharing through pairing – the designer benefitted from picking up skills from the development process. In particular, he started working with the coding environment so that they could both collaborate on the same codebase.

Another success was to very quickly become familiar with Git, and also PHP frameworks and the rest. It felt a bit like being dropped in the deep end, in a good way.

Being closer to the medium for which we're designing felt more 'right' to me and I'm keen to work like this again in future.

Theme Category 2: Increased efficiency and iteration of ideas

(i) The study showed that by working together in this way made it quicker to create solutions that would have previously taken a lot more time using the usual process. In particular, working in this way allowed the pair to learn about the potential problems of their solutions straight away so little time was spent pursuing over-complex ideas.

It was incredibly productive and I learnt things about the quality of some ideas within minutes rather than months.

We have created in a week what would normally take months and can immediately spot the problems rather than inflict them on our audience.

It's set us up to be more efficient too, partly because it's raised a tonne of difficult questions for us to start solving straight away - and partly because we have the tools, the process, and the prototype itself to build upon.

(ii) The pair found that the working closely together in this way helped them to be more productive and create better, more robust solutions. They both believed that was a model or a way of working that should be applied more often in ASD teams.

We have a working model that can be perfectly suited to each participant, without any extra work.

7.4.4 Prototyping and delivery as a pair

Theme Category 1: Collaborating to produce prototypes with realistic data

(i) The pair found a lot of value in working together to produce prototypes because they represent the design concepts in a realistic way, helping to improve their decision-making. From producing the prototypes, they also had something that was closer to production if the concepts were taken forward.

> A new process [for the designer] was to code up layouts with 'dummy' content and then to pass it to [developer] to plug in with real data - this felt like a really efficient way of working, even if our questionable front-end code needs tidying up before anything makes it to production.
>
> For our own internal purposes, working with real data, real images etc. has brought some of our visual decisions into question which can only be a positive thing, since we're identifying those issues earlier.

(ii) By creating prototypes it gave people from the wider team the opportunity to provide opinions and give feedback, allowing the pair to learn and iterate on their work. Feedback like this could of occurred in their previous process but the advent of a prototype that uses real data made their work have more gravitas with their colleagues, it was "real".

> This prototype has already done the rounds with stakeholders and getting great feedback (as well as stimulating negative comments which are more useful to hear early on).
>
> There's something about making our concepts 'real' which raises stronger opinions, and questions, and generally seems to have (hopefully) sped up the design process.

Theme Category 2: The wider team has different expectations of delivery between the pair

(i) Having prototypes as outcomes of the study gave the opportunity for stronger opinions from other members of the team and an expectancy about the output that the pair did not anticipate. Because what they had created felt so "real" to people outside of the pair, they believed that it was close to production when in fact it was a prototype.

> I would say the major problem has not been the way we worked together but the expectations around the output.
>
> People (stakeholders) assume that since something has been built it will be amazing and they will be able to put it live next week. This goes back to my point about the 2 sets being perceived as different.

> If designers spent a week together thrashing out some ideas the expectation of output would be completely different, certainly no one would expect anything to be built.
>
> Two 'fears' that I have are firstly that stakeholders see this as a 'nearly final' design that is ready to build, and secondly that future design decisions may take longer if it becomes expected that everything needs prototyping to this level of fidelity.

The qualitative analysis of the comments in the questionnaire after the pairing study reflect some of the perceived benefits of close collaboration and pairing that emerged from Study 1 & 2. The outcomes also echo what the pair perceived would be beneficial before the study took place.

The findings show that the pair enjoyed more frequent side-by-side communication and that by working together it was quicker to create solutions that could then inform decisions about how to move forward. These solutions were thought to be improved from what was normally produced because they were functional prototypes using "real" data.

The novel outcome that was found from this study came from the shared production of prototypes, as opposed working with "flat mock-ups" and specification files. The pair believed that prototypes would be a useful solution from working together but through their work they proved this out. They gave the opportunity for user testing and brought about stronger opinions from other members of the team and wider stakeholders. The prototypes were also perceived to be closer to something that could be released so members of the team took them more seriously.

The production of the prototypes in a short space of time raised questions about the expectancy of the outcomes of a design and development pairing and how an iterative process can be developed in line with stakeholder and team expectations in the ASD process.

7.5 Discussion of Study 3

The current study focused on finding out about the collaborative relationship between a designer and a developer who were paired together for 1 week within the context of an ASD product team in a large media organisation.

From the findings of a questionnaires before and after the week of pairing, a picture emerges that confirms some of the findings in Study 1 & 2, whilst also providing qualitative evidence towards successful factors that pairing enables, including improved collaboration and communication between the two roles, creating more joined-up thinking and knowledge sharing to produce prototypes with "real" data together that the wider team could then make decisions upon.

The previous section described the findings from the questionnaires to provide a picture of how UCD integrates with ASD in an organisational setting and how pairing over the course of 1 week can make an impact upon the collaboration between designers and developers. This next section discusses these findings to understand how they relate to the literature and the research questions outlined in this study so that the next steps can be formed.

7.5.1 Organisational separation hinders collaboration and knowledge sharing

Before the pairing period was initiated in this study, the pair described the typical working circumstances in their team(s) as an on going challenge for collaboration and knowledge sharing. Both roles perceived that their organisational structure hindered their ability to work closely together on a regular basis. This was thought to be because of the separation in their teams where they were placed in structurally different divisions within the organisation.

This separation, both in terms of physical location and via 'invisible' team structures was felt to lead to a lack of regular overlaps where knowledge and practices can be shared and joined-up. This reduced the crossover of skills and awareness of their work and correlates with Sy and Miller (2008), who describe that non co-location of the roles results in collaboration barriers and communication difficulties. In this case, the organisational culture drives the separation and non co-location of the two roles, confirming findings by Ferreira et al. (2011) relies heavily on the designers and developers (1) mutual awareness of each other where they

must step outside of their immediate teams to (2) engage with each other to achieve integration in practice (Ferreira et al., 2012a). Despite this, the designer and developer in the case were not engaging effectively with each other on a regular basis although they were aware of each other.

Although this study is the case of just two participants in the organisation and is not readily generalisable, this further confirms findings from Study 1 and 2, where a picture has formed in regards to the organisational setting; the separation of the UX team(s) from the Agile development team(s), making it an on going challenge for UX designers and software developers to regularly come together for collaboration and knowledge sharing.

This organisational separation between teams was an active source of frustration for both participants who were keen to be more joined-up and collaborative. In particular, by working side by side through 'pairing' they believed their work would be more efficient with a higher degree of knowledge sharing, allowing them to create a better overall experience with more crossovers and less up-front design work.

7.5.2 Inefficient "handovers" of design documentation

Separation and a lack of close collaboration between the two roles in the organisation is demonstrated with a heavy handover process from the designer to the developer. This issue has been highlighted in the previous studies and it is a conscious issue between the two participants in Study 3. In particular, it was felt that design work was perceived by both roles to be happening too much in a plan-driven way where the work is handed over to the developers via "specifications", "flat-mockups" or "mark-up" files. This was found to be inefficient and time consuming for the developer to retrospectively understand the design details with an absence of the designer from their day-to-day work and caused frustration. This contradicts with the UCD and ASD principles of facilitating iterative design and development with reduced documentation and adds to the reported controversy of how much 'upfront design' is appropriate (Adikari et al., 2009). This also fits with previous observations for a need to frequently 're-align' work processes and product development plans (Brown et al., 2011) and that integration of ASD and UX relies on frequent negotiation between these roles (Ferreira et al., 2012).

Additionally, the designer in the study voiced issues about finding it difficult to complete their work and collaborate with developers because of the tooling available for designers. In particular, it was felt that it is difficult without the specific knowledge or skills to become integrated with development processes. The difficultly in sufficient tooling like this was reported by Coatta and Rutter (2011) and due to the separated nature of the two roles in the organisation, this problem could be compounded and add to communication problems, making up-front handovers slower and more problematic than they should be.

Frustrated by the these difficulties, there was a desire between both roles for the design and development process to be more aligned to UCD and ASD principles by being more iterative in their approach and reduce documentation. In particular, by reducing upfront design work, they hoped to be more flexible in their approach early on the process. By pairing together they believed that what they create could be more of a shared process involving the production of more prototypes to act as a talking point rather than a "flat mock-up" of the experience.

7.5.3 Better knowledge and skill sharing through pairing

In this study, after the pairing week, a picture emerges that shows that pairing between the two roles improves knowledge and skill sharing due to being located side-by-side and benefitting from more frequent, ad-hoc communication. This collective sharing was valued because it enabled the mixing of their skills to improve their work. Examples included the designer finding a greater understanding of performance of the site and the design implications upon the user experience were enhanced with this in mind.

The knowledge sharing and crossovers between the two roles was improved through pairing and collaborating together on the same project from the start. This made it quicker to create solutions that would have previously taken a lot more time using their usual process. In particular, working in this way allowed the pair to learn about the potential problems with their ideas straight away so little time was spent pursuing over-complex ideas. Through pairing, the designer also benefitted from picking up skills from the development process. In particular, they started working with a coding environment together so that they could both collaborate on the same codebase.

Although these benefits are promising and appear to improve their experiences that were felt prior to pairing there are implications. As a pair in the organisation, they had been given this project for 1 week and it was different to any other project they were currently involved in, without any of their surrounding team members or other commitments. Naturally, by bringing the designer and developer together to sit alongside each other in this way, it is expected that previous ways of working would change and the roles to be more joined-up, with frequent communication and adapting their style to be more collaborative. Despite this, none of the collaborative practices that were demonstrated in this study were being permitted from taking place during their normal working circumstances.

7.5.4 Co-creation of prototypes with "real" data

A key finding that was identified in this study was the shared creation of prototypes between the design and development pair. The participants emphasised the creation of the prototypes as a shared output of their work which they were both invested in. This contrasts with the literature where it is reported that designers should be the ones to create prototypes and according to Chamberlain and Sharp (2006), they should be willing to "feed the developers" with prototypes.

The pair used these prototypes with the wider team and stakeholders to help make decisions on how to move forward. They also planned for the prototype to be used to test the experience with users. This is a dramatic shift from their aforementioned process prior to the pairing work where heavy handovers of mock-ups would be used evoke conversations with the wider teams after a period of upfront design work. The change in output, from a flat mock-up to a working prototype with real data – created a shift in perceptions from the surrounding team as getting to a working prototype normally would take a long time and usually would be produced solely by the designer(s).

The pair both found a lot of value in working together to produce prototypes because they represent the design concepts in a realistic way and it helping to improve their decision-making. By producing the prototypes, they also had something that was closer to production if the concepts were taken forward following discussions with their wider team and testing with users. For example, by creating prototypes it gave people from the wider team the opportunity to provide opinions and give feedback, allowing the pair to learn and iterate on their work. Feedback like this could of occurred in their previous process but the advent of a

prototype that used real data made their work have more gravitas with their colleagues because it was “real”. The negative aspect of this was the expectancy about the output that the pair did not anticipate. Because what they had created felt so “real” to people outside of the pair, they believed that it was close to production when in fact it was a prototype.

7.6 Summary

In summary, as with the Studies 1 and 2 this study continues in the vein of casting the integration of UCD and ASD in practice as a problem situated in the organisational setting in which the UX designers and software developers are structurally separated into different teams. With organisational support, for this study the UX designer and software developer were seated together for a period of 1 week as part of an ‘innovation period’ where they exhibited a higher degree of collaboration through frequent pairing together.

Their closer collaboration resulted in a quicker output, improved knowledge and skill sharing and a collaborative process that resulted in the co-creation of working prototypes that could be used by the wider team to make key decisions about how to move forward.

However, although the discovery of making prototypes in this way was interesting, one key aspect of this study was that it took place over a just one week and that it did not necessarily truly represent the ebb and flow and the naturalistic process normally employed by the participants and their wider team in an ASD environment.

So whilst collaboration was much improved involving the successful factors described, it is not possible to say that this model of design and development pairing would scale beyond this context and setting within different types of projects and scenarios.

7.6.1 Next steps

So with the implications of this study in mind, and to understand how this model might scale, the next step for the research was to apply the pairing approach to a broader range of participants involving more teams, circumstances, contexts and potential variables across a diverse and varied organisation.

Chapter 8
Study 4 - Pairing Study B

8.1 Introduction

The chapter presents an ethnographically informed study with six design and development pairs to learn about how they collaborate and work together over the course of six weeks.

The aim of the study was to understand the determining factors for successful collaboration between design and development pairs. To do this a Contextual Inquiry with each pair took place to capture naturalistic insights. A Longitudinal Study was also undertaken using individual diary studies to gain further understanding.

The next section (8.2) presents background information about the factors involved in this study, including the context, the participants, the projects they were working on and their organisational setting. Next, the data collection and analysis methods are discussed (8.3). Next, the findings that emerged from the analysis are then described (8.4) and then these findings are discussed. Finally the key findings of this study are then summarised (8.6).

8.2 Background

As discussed in Study 1 & 2, both roles and their stakeholders perceived that collaboration could be improved. In particular, co-location and pairing of designers and developers was desired as an opportunity to enhance more localised and autonomous decision-making, especially early on in projects. Motivated by this finding and the outcomes from Study 3, this study further explores how a collaborative pairing approach between the two roles could improve the process of integrating the two roles and the processes of UCD and Agile. To do this 6 design and development pairs were established across different teams and a Contextual Inquiry which each pair took place. In addition a Diary Study over the course of 6 weeks was undertaken to gain further understanding.

The study took place within the same organisation as described in Study 1, 2 & 3, a large media organisation in the UK that develops digital products for large diverse audiences and typically adopts a varying degree of ASD processes across their product teams and locations. Each of the pairs was setup across a different online product team. The offices were in different locations but all had an open plan space with banks of desks, break-out-areas for ad-hoc meetings and collaboration and meeting rooms that are available to book out. The surrounding teams varied across the participant pairs but all pairs were part of a multidisciplinary product structure that included Designers, Developers, Business Analysts, Testers, Technical Architects, Project Managers and Product Managers.

From the 5 previous studies conducted in this thesis it shows that the two roles, and their wider stakeholders (line-managers), want to bridge the gap in collaboration, with the aim of improving knowledge sharing, motivation, efficiency, decision-making and the quality of their output, particularly through the use of prototyping and using user data to inform.

From the initial positive outcomes of pairing in Studies 2 and 3 the next step is to investigate the pairing model further and to find out how the themes listed above can provide insight into how several Design and Development pairs find success in their every day work.

8.2.1 Participants

In terms of their specific roles, the Designers were classed, as "User-Experience Designers" and the Developers were mainly "Front-End Developers" or "Web Developers'
To select participants stakeholders (line-managers) from around the organisation were contacted and asked to suggest a range of individuals. From this list of eligible participants the method previously used, known as purposive sampling (Palys, 2008) was applied followed by contacting the potential participants to take part. Participants had the permission from their line-managers to take part in the study. This gave each participant reassurance and an understanding from their wider team that they were involved in a research study.

Out of the six pairs of participants, pairs A (A:Des, A:Dev) and B (B:Des, B:Dev) knew each other already, they had worked on projects together and were part of the same Agile team. Pairs C (C:Des, C:Dev) and D (D:Des, D:Des) knew each other, had worked together before on a project but were not in the same Agile team. Finally, Pairs E (E:Des, E:Des) and F

(F:Des, F:Des) did not know each other particularly well and they were structurally placed within separate teams but within in the same broader product team.

8.2.2 Projects

Their projects varied across the pairs but these were comparably concerned with designing and developing web-based content. It helped to form a picture of what its like for designers and developers at many difference phases of a product development cycle and how this impacts upon their collaborative practices together. In this study all of the projects varied in terms of their aims, size and scope across the six pairs of participants but the consistency was that they were all working on web-based digital products.

The designers A:Des and D:Des were working on a one main project between them whilst the other designers were all spread across various projects across the product or the company wide UX team. Whereas the developers were all attached to a single main project within each product. The organisation would often have hack days or technical spikes but in this case all of the projects were classed as on-going product development projects meaning that all of the pairs were working as part of their typical product constraints and timeframes. Further details about the situational context of the projects and the participants for the six of pairs is described in Appendix C.

8.3 Data Collection & Analysis

For this study, it was important to capture data about how the participants worked and how they perceived their practices in a naturalistic setting. To do this, the study design needed to allow for observations, semi-structured interviews and periodical accounts of their work.

Therefore, the data collection was split into two parts: a Contextual Inquiry in the form of semi-structured interviews at the very beginning of the study, and a longitudinal study in the form of a participant diary study over the course of six weeks. Observations were taken down as notes and audio recordings were transcribed following each session with the participants.

Diary entries were captured using a blogging tool called Tumblr[9] which allowed each participant to submit entries online or via email.

Each individual was asked to take part in the pairing study and they were briefed about what would be involved along with filling out a participant information sheet. Context was provided about the aims of the research and why they were being observed and studied. The factors involved in each part of the research (CI and Diary) were provided for the individuals and are detailed below.

8.3.1 Contextual Inquiry

A Contextual Inquiry (CI) is a semi-structured interview method used to obtain rich information about work practices, the social, technical, and physical environments, and user tools (Wixon et al., 1990). Participants are first asked a set of pre-defined questions and then observed and questioned while they work in their own environments. This method that can be adapted to suit different situations and goals. The interviews and observations take place in the participant's normal working environment where they can explain their typical processes, tasks, environment, opinions, tools and their communication patterns. A CI defines four principles to guide the interaction:

Context—Interviews are conducted in the participant's actual workplace. The researcher observes participants do their own work tasks and discusses any artefacts they generate or use with them. In addition, the researcher gathers detailed stories of specific past events relevant to the project focus. If specific tasks are important, the user may be asked to perform those tasks.

Partnership—Participant and researcher collaborate to understand the participant's work. The interview alternates between observing the participant as they work and discussing what they did and why.

Interpretation—The researcher shares their interpretations and insights with the participant during the interview. The participant may expand or correct the researcher's understanding.

[9] https://www.tumblr.com/

Focus—The researcher steers the interaction towards topics, which are relevant to the project scope.

The main benefit of CI is that participants are interviewed in their own environments, so this type of research can capture more salient, nuanced behaviour and work-environment relationships than standard interviews or user tests.

In this case, the focus of the interviews with the designers and developers was to analyse what has and has not worked in the past. It was stressed that the interview results would be kept as anonymous as possible. As part of the inquiry, interview sessions with the pairs took place as well as the individual sessions. The focus of the meetings with the pairs was to openly discuss their previous experiences and their requirements and expectations for their future projects. The aim is that participants speak freely and talk about issues and barriers to success. Table 8—1 below shows an example of the scripts that were used for the contextual interviews with the design and development pairs. Importantly, these questions did not aim to act as an exhaustive script, but instead were there to guide the flow of the conversation with the participant(s).

Context	**Questions**
Paired interview	Which projects have been the most successful and why do they think that was? What are the strengths and weaknesses of the current collaborative methods? Does ASD work well with UCD? How do the UX designers adapt what they do to fit in within the ASD process? How would they ideally do it if they could do things differently and why? What would they change about the current process? Do they know why collaboration might not be working? Do they have any examples? Is there currently a defined process? Should there be?
Individual interviews	What is their previous experience of agile and working closely with UX/development teams? What three things could the UX/development teams do to make their job easier and collaboration more successful? What do they like / dislike about working in an agile way? What do they think are the strengths and weaknesses of working as part of a collaborative partnership?

	Do they think they will like working as a collaborative pair? What are their expectations of their new partnership? How would they like to work?

Table 8—1: Contextual Interview Sessions – Discussion Guide Questions

8.3.2 Diary Study

To capture experiences over a period of time as projects changed and developed, longitudinal data was collected via the use of online diaries. Each participant was asked to report over the course of 6 weeks after participating in the CI. This was broken up into three key parts; the participant briefing, their on-going diary entries and follow-up communications by posting questions to them online. The methods for these parts are detailed below. Participants were briefed during the CI. In addition, participants were sent an email with essential information to take part in the study. They were informed that they could contact the researcher by phone or email if anything needed to be discussed or should any problem arise.

The designers and developers were asked to operate as pairs and collaborate together in their work whenever they felt it was appropriate. A briefing document with the details for the diary entry template and examples was also provided. Each participant was also asked to make an initial entry at the very beginning of this phase to ensure clarity about the process. They were asked to upload a photograph of themselves along with a very short bio. The entries were collated on a Tumblr[10] blog (which allowed entries via e-mail, smart phone or the blog website) that was password protected for each of them. Participants were asked to submit pictures taken from their smart phones at work, videos etc. throughout the study. This could be separately with a brief explanation or tag, or within an e-mail diary entry. The researcher would then follow-up on interesting or incomplete diary reports via e-mail. Participant entries were monitored closely during the first two weeks to ensure participants are providing quality entries with sufficient detail and understand the aim of the research. During the following days, the researcher reviewed the entries and prompted or guided participants with questions related to research goals that have not yet been addressed.

[10] https://www.tumblr.com/

Efforts were made to ensure that every participant was engaged during this period. The analysis was to be applied to the data available at the end of the study, even if some participants dropped out.

8.3.3 Analysis

As described in section 8.2.1, participants had different levels of pairing-experience going into the study. The nature of their work and the projects they worked on was understandably very different. The diversity and structure of their wider teams varied and this impacted the projects and the tasks the pairs worked on. Although the researcher observed these factors, the main focus of the study was on the interactions between the designers and developers. Aiming to understand the determining factors for successful collaboration between the pairs and how it could work across different circumstances and project tasks.

The transcripts and observations from the Contextual Inquiry and the Diary Study data were analysed separately using a thematic content analysis method. Inductive codes are derived from the data as the sorting and analysis occurs. Expressions of opinions, problems, events, reactions and interactions in the text are coded by assigning the piece of text to a category (''node'' in NVivo). A category represents a phenomenon, that is, a problem, an issue or an event that is defined as being significant. When categories were found to be conceptually similar in nature they were grouped under more abstract, higher-order categories.

Following the analysis of the Contextual Inquiry and Diary Study, the themes from both data sets were analysed together using a triangulation approach. Data triangulation uses different sources of information in order to increase the validity of a study. By triangulating the analysis of the two data sources it increases the "confidence in research data, creating innovative ways of understanding a phenomenon, revealing unique findings, challenging or integrating theories, and providing a clearer understanding of the problem" (Thurmond, 2001, p. 254).

8.4 Findings

The findings in this section are presented as follows: Firstly, the findings from the thematic analysis of the Contextual Inquiry & Diary Study are presented in Table 8—2 and Table 8—3 below. The resulting themes from the triangulation of the two data sets are then described with a selection of supporting evidence and finally are then summarised Table 8—4.

Theme Categories	**Team collaboration & integration challenges**	**Pairing experiences and closer collaboration**	**Further opportunities for better integration**
Pairing	*A "perceived" separation between the two roles based on workflow and goals*	*Collaborating to create prototypes and in-browser style guides and pattern libraries*	*Desire from both roles to use more data and audience research together to inform their decision-making*
	Separation of location caused frustration and led to less efficient ways of working	*Direct and frequent communication throughout a project sets expectations and helps to build relationships*	
	Lack of direct communication between the roles	*Early ideation and experimentation together helps to share ideas and define the increments of work*	
Team	*A lack of shared understanding of each other's discipline*	*Breaking down tasks to make small decisions and iterations together*	*Desire to be together as a pair and multi-disciplinary team where roles can crossover*
	Both roles perceived that team size was too big and the structure was complex	*Location and close proximity is an enabling factor for close collaboration*	
Organisation	*Frustrations of a lack of shared decision-making in their team*		*Positive experience of Agile and a desire to be more true to using Agile Methods*

Table 8—2: Contextual Inquiry – Table of thematic analysis findings

The diary study took place over the course of 6 weeks for the 6 pairs and their individual diary entries have been analysed qualitatively with the same approach as the Contextual Inquiry. The findings from the analysis are presented below in Table 8—3.

Theme Categories	**Team collaboration & integration challenges**	**Pairing experiences and closer collaboration**	**Further opportunities for better integration**
Pairing	*On going separation between UX designers and Agile Developers*	*Pairing enables efficient and iterative collaboration to find solutions to work*	*Want more opportunities to sit in close proximity with each other*
		Location and close proximity is an enabling factor for close collaboration	*Desire for more pairing and closer collaboration*
Team	*Too many stakeholders and layers to the team structure*	*Helping to breaking down 'invisible' barriers between roles and learn skills from each other*	*Challenges of extending pairing method to more teams*
		Paired work is seen to benefit whole process	
Organisation	*On going separation between UX designers and Agile Developers*	*Wider decision-making is improved and more informed*	

Table 8—3: Diary Study – Table of thematic analysis findings

Following the analyses of the Contextual Inquiry and Diary Study, the data triangulation method was applied. The subsumed themes are described in the following three subsections: Team collaboration & integration challenges, Experiences of pairing & closer collaboration and Further opportunities for better integration. The themes from the study are then summarised in Table 8—4.

8.4.1 Theme Category 1: Team collaboration & integration challenges

This section describes the collaboration & integration challenges that were experienced by the pairs of designers and developers.

Theme Category 1.1: Perception of operational separation between the two roles

(i) All pairs observed a situation of operational separation between the two roles - a "wall between UX and Dev" – that would create problems in terms of quality and efficiency and often would result in "more bugs going live".

Pairs C to F perceived that there was a palpable sense of separation or a "gap" in workflow and interests regarding the outcomes between the roles. This would then typically slow down the workflow and efficiency because designers' work was perceived to start and finish before developers would get involved. Pairs A and B, who began the study having already worked in pairs, also confirmed this separation through their recent past experiences as part of their wider teams.

> One of the initial issues that stops this kind of work is that Designers and Developers are often in separate teams to begin with and also people might have worked in this way for a long time so its engrained in their behaviour.
>
> People are still stuck in that mentality of 'they' design and 'we' develop, and we clash every now and then.
>
> So, there are big barriers between designing something and the development process.

(ii) This separation between the two roles was also reflected by the developer's frustration with the handing over of mocked up "flat designs" that do not suit their requirements. Problems would arise – according to comments - because the design was often created without the understanding of the contingencies or the knowledge of the constraints and understanding the data.

> We seem to have a very rigid way of working where UX supply the designs, the developers build them, and we launch - there isn't much room for agile working to make changes on the fly.
>
> Designers bring us pretty Illustrator flat designs for screens.
>
> But it's more the case of 'here is the design, go away and build it'.

(iii) Developers felt that working with designers is not flexible enough (e.g., not enough ad-hoc meetings) and a lot of planning is needed to finish release cycles. For example, the variations of a component would be difficult to predict in the designs and would be delivered without the regular input from designers. This would cause blame and tension between the two roles when the delivery of the UX would not live up to the designer's vision. This was most common in pairs C to F but pair A also cited this problem as an issue within their wider team.

> Without that understanding, this can cause quite a lot conflicts within the team because people are saying, 'right I've designed it to have 24-pixel margin, I want it to have this' but that's not necessarily the case of how things are built.

> Designer just does the component but don't test contingencies.

(iv) The issue of separation is amplified because the work priorities of designers are perceived to be at least partly different to those of developers. Consequently, frustrations often arise about UX tasks and goals being seen as de-prioritised over other work, accompanied by a sense of a lack of progress regarding the design work.

This issue is often perceived to be due to the two roles not communicating enough ad hoc with each other and working on different projects at the same time. All pairs mentioned this but most adamantly by Pair F who were not at all aware of what their counterpart was working on, which resulted in a combination of a lack of knowledge from the designer and a lack of understanding about priorities from the developer.

> UX tickets don't get a review or are constantly de-prioritized.

> Developers do not fix visual/UX issues quickly because other 'real' bugs have priority.

Theme Category 1.2: Lack of direct communication between the roles

(i) Participants from both roles mentioned problems that arose from a lack of direct communication early on in a project. The pairs reported frustrations in previous projects that had resulted in problems because communication was not present early on in the project. Despite Pair A showing that they communicated well with each other they thought that communication between other designers and developers was "lacking" and could be improved. With the exception of Pairs A and B, this was shown through a lack of day-to-day communication between all other pairs.

> Communication between UX and Dev is lacking.

> Not contact or communication with development team unless when working as part of a product team.

> But of course, one problem is separating from each other […]. They cannot discuss their issues.

(ii) The roles were often separated from each other in either location or organisationally because of team structure. It meant that they could not regularly discuss ideas or issues with together which resulted in the overall lack of communication. Problems would then arise during the process, causing frustration for both roles.

> It's just starting a dialogue as early as possible, so you never get to these situations in the first place.

> This wouldn't happen if UX came earlier to us, captured earlier on in design process, if they came to us developers to talk about it.

(iii) When electronic communication was used to collaborate between roles it was perceived to be cumbersome and inefficient. It was found that the pairs had learnt to try and avoid it due to previous problems.

Through e-mail it loses things […] Electronic communication just doesn't work as well.

It's not the same quality of feedback, you have to plan around that.

Theme Category 1.3: A lack of shared understanding of each other's discipline

(i) For the designers, experiences from previous projects had affected their perception of working with developers. A frequent type of comment was that the language and jargon developers were using is generally hard to follow and often designers therefore felt they cannot contribute to discussions and meetings (e.g., 'stand-ups' and 'sprint planning').

I understand only 80% in stand up.

Often, I get lost in the jargon (Dev and Agile jargon) and abbreviations.

I also had a very annoying conversation with a developer who asked for a chat about a ticket he'd picked up regarding inconsistent spacing. I was very confused about what he was saying and his reason for not being able to make the fix until another developer pointed out he had completely misunderstood the ticket. Not sure how this happened!

(ii) At the same time, developers showed a desire to be more involved in the design process and if possible to learn more about UX and the UCD process. In particular, developers wanted to have more input and feedback from users but most changes (e.g. to well established features) seemed to cause negative feedback from the audience so this caused ambivalence to how the process works.

Would be good to see what they [designers] are thinking!

Also, for developers would be good to have a bit more UX training.

(iii) Designers spending time to learn about key aspects of the development process were perceived as beneficial to the overall success of the project. Developers mentioned that if designers up-skilled themselves or received training in aspects of coding or the software engineering development process it would be beneficial as a better understanding of timings and constraints would be gained.

This training had occurred in Pair D and had benefitted the process from the developer's perspective in saving investment upfront. In Pair A, the designer had also spent several weeks learning about the development process with assistance from the developer, in particular, learning to use Github (a code sharing and collaboration tool) which enabled them to collaborative more frequently together.

Some of the designers come from a web-background and has helped them more.

Often designers coming from print background don't understand technicalities of web design […] I had to make a case why it wouldn't work, and they were heavily invested in it […] UX people are a bit more knowledgeable now with the training.

I think it helps you to understand not only how we build things but how we release things to the public and what that process is and why it's like that. […]. Without that understanding, this can cause quite a lot conflicts within the team.

Theme Category 1.4: Separation of location and a lack of proximity

(i) Location - in terms of working in the same building, floor, or area - was perceived by the pairs to be an important factor for successful collaboration. When the roles were separated, even by just a few meters, it could detrimentally affect the working partnership between the pair.

The majority of the pairs were working in separate locations before the study. At first, only Pair A and B were already sitting alongside each other and they both found benefits from this situation, especially having experienced the difficulties in previous work when they worked in a different location to other disciplines. Pairs C and D sat on the same floor, several meters apart, and met for ad-hoc sessions. Pair E was separated by different floors but in the same building. Lastly, Pair F was on the same floor, but separated by a few team areas and they reported little interaction with one another. All pairs reported that being in separate locations was unsuitable to close collaboration (despite Agile routines and remote communication). This was often out of their hands due to lack of space or the structure of their surrounding teams.

> Quick small tweaks less likely when we sit further away from each other […]. That would not work for me at all. You are less inclined and go up and ask them.

> Yeah, this was the problem and I was sat here you were sat at the other end of the building, just wasn't working, that's why [the developer] needs to sit next me and we just need to do it together.

(ii) The seating arrangements of the designers and developers did not have flexibility to suit the situational nature of collaboration between them and their colleagues. Pairs C, D and E expressed frustration around the location of themselves and their colleagues and how it would be "a blocker" in collaborating effectively, they had all expressed positive examples of working in close proximity with the other role, but this had often not lasted long, and change would be out of their control.

> At one point we all sat together, it was good for a while […] guys would come over and ask. But then they split the dev team into two work streams. There wasn't space for me to sit here anymore and the relationship became more difficult.

> Sitting together was a big thing. Location seems like a small thing, but it makes a huge difference. The team seemed quite good at taking to each other. Explicit collaboration is difficult.

(iii) Additionally, the location difficulty was often thought to stem from a lack of general space in the working environment for ad-hoc meetings, discussion and collaboration to suit the needs at the time.

> There is no seating, lack of space, generally no room to sit.

> Need more quiet areas […] Overbooked, and a quiet space ad-hoc; you just want one now.

> There is no seating, lack of space […]. Need more quiet areas but they are difficult to book.

Theme Category 1.5: Frustrations of a lack of shared decision-making

(i) It was found across the different teams that there were too many people involved in the contribution to the product, meetings and part of the decision-making process. The nature of the large organisation requires communication and collaboration to occur across teams for different purpose, hampering the localised level of decision-making and autonomy of the pairs and their teams.

> I had a chat with a designer from another team today who echoed my thoughts that there are too many managers involved in projects at the [organisation] and it hampers out ability to work in a lean and agile way. There needs to be a change from the ground up to allow us to work more effective and efficient way.
>
> Today has mainly consisted of working group meetings that didn't really get anyone anywhere. There was nearly 20 people involved in one of them and it seemed like a waste of everyone's time.
>
> Whatever I do here there seems to be far too many people involved and this seems to be a major contributing factor to why it takes so long to get anything done here!

(ii) Designers and developers showed general frustrations about the lack of involvement from their stakeholders e.g. their creative director or product owner. As there was an organisational requirement for them to sign off and make key decisions about the work this lack of involvement meant a slowing of progress and a feeling of a lack of ownership. Consequently, there was a desire for more localised decision-making within the teams.

> He [the creative director] would turn up and say do this and this, so far removed from day to day and doesn't know about our pressures, without sitting down and talking to us.
>
> The decision was made by him [the general manager] and gradually filtered down to us […] to get sign-off on something can be incredibly difficult.
>
> Stakeholders and critical decision-makers are not available enough.
>
> If I want to get sign off over something more significant, I will go to the product owner and ask if it works and if it's not perfect, can we live with this for now? The problem with this approach is that it can stall progress.
>
> There is a lack of shared decision-making in team.

(iii) Despite the frustrations requiring sign-off from stakeholders on certain decisions, so called "small" decisions (about relatively minor design issues) were found to often happen via agile based collaborative 'review' methods or user testing methods in their disciplinary teams.

However, this was still restricted to each other's discipline and there was little evidence of crossover in this sort of decision-making. For developers this would occur with group discussions such as agile ceremonial 'reviews', stand ups and via electronic methods known as 'pull requests'. These 'pull requests' were particularly common for the developers, who mentioned that they were used to manage the changes in the code collaboratively with other developers in the team. For designers,

decisions would often occur via ‘design critiques’ in their design teams or via user testing methods such as usability lab testing or guerrilla testing.

> [On working with other developers] We work on our own or in pairs, depending on tasks. I do my code changing, then add them to Github and then ask someone to look at it. Then it gets approved and goes to the mainstream Github branch.

> [On working with other developers via pull requests] Yeah so you can assign it anybody in the team, you choose who and usually someone who has the knowledge of what you're working on and sometimes it might have 6 or 7 people working on it. You can comment on any line in here, this might need changing or it might be really good.

> [On working with other designers to make decisions] I also have team crits and design crits with the UX team, so we can work out which stuff is working and have an idea of how the UX is spanning across the product and what can feed into other areas or projects and if they have any design recommendations or things like that.

8.4.2 Theme Category 2: Experiences of pairing & closer collaboration

Designated pairing of a developer and a designer and close collaboration between the roles provided improvement examples that have been described in their relevant themes listed below.

Theme Category 2.1: Working in close proximity

(i) Location and close proximity is an enabling factor for close collaboration. Close proximity in terms of location with the other role was seen as particularly important to enable side-by-side communication and regular ad-hoc discussions.

Co-location was perceived as a key factor in being able to make iterations quickly together and communicate and provide feedback frequently. This was found most commonly observed between Pairs A and B but Pairs C, D and E also acknowledged that their previous experience of sitting close-by to the other role in the past had enabled much closer collaboration and frequent communication.

> Sitting close to devs is good to resolve small problems so a dev makes a change in the code and you can see right away how it affects the design.

> I have been in teams where designers sit next to Dev […] And that really work very well […] You need the fast feedback.

(ii) By sitting in close proximity to one another, either alongside each other or on nearby desks, it enabled more frequent communication and closer collaboration.

> Today we moved back to sit with the developers. They have been asking lots of questions - this is good! It makes life A LOT easier!!!

> Communication is made easier when Dev and UX sit close by. I always prefer face - to - face communication over anything else.

> After exploring some visual tweaks, myself and [name] have realised some of these tweaks may not be possible (table cell border) - down to the way these components are developed. This is the real benefit of sitting together and just generally chatting through our design thinking throughout the day.
>
> After about 3 months I moved round to sit near the developers, and our work stream steadily improved and everybody felt more integrated.

(iii) When the individuals were on nearby desks but not directly next to one another it was still found to be useful for ad-hoc discussions and stand-ups.

> Being the only UX [designer] with all Devs [developers] felt a bit strange whereas now we are all UX and it feels more like we are UX team but also, we are close to the Devs and we are close enough and we have daily stand ups.
>
> Sitting close to but not necessarily next to developers appears to work really well, encouraging a balance of creativity, teamwork and strong communication.

Theme Category 2.2: Early and frequent communication

(i) Direct communication early on in projects helped to reduce problems later on because of relationship-building and early sharing of ideas. It was found with Pair A that because they had been communicating closely together since the beginning of the project they both perceived that it had helped to reduce problems later on.

Direct communication in person was preferred amongst the pairs because it would enable discussions and ultimately solutions to the particular work in question. Pair A in particular believed that by frequent verbal exchanges they eventually ended up with the better solutions, as face-to-face communication allowed them to iterate their ideas more easily and more frequently. Face-to-face communication like this was common between pairs A, B and D and didn't occur as frequently in the other groups but all of the pairs perceived it to be the best way to communicate to share ideas or solve problems. Communication in person also occurred in groups through the use of agile ceremonies (e.g. 'Three Amigos', 'Retros') to help have regular discussions.

> It's just starting a dialogue as early as possible, so you never get to these situations in the first place […] you can address problems really quickly rather than having a conflict down the line when its more critical.
>
> [Name] has done a bit of sketching work on Illustrator, but we talk all day long. And I say for example, I don't think that's going to work, or something is breaking the design and we say yes, we have to move it. Just talking and collaborating, but actually doing it on the browser.

(ii) Pairs observed that by having regular communication it had helped to develop their relationship and a mutual understanding of their roles. The Designer in Pair B was really positive about the effect of this - "it feels like a family […] with lots of banter". Pair D also acknowledged that if they communicated more, and thus developed a stronger relationship between each other, the team would be also being stronger.

> We would have a stronger team if we had more of a relationship.
>
> Getting the UX – Dev working relationship very strong is important.

(iii) When the pairs were not in close proximity of each other, electronic communication was often used to assign work between roles and individuals in the team.

With the exception of Pair A, it was found across the pairs that electronic communication and tools were used to assign work to one each other and other people within the wider team. This was often due to location differences between the pair or team members. Pair A would only assign each other work by collaborating and sharing out responsibility whilst sitting next to each other.

> Our process is that we will assign it to him (the designer) in JIRA... and then if it's not picked up we'll then mention it in stand-up and often then sort it in person so it's not a massive problem, we are generally only waiting sort of like... 12 hours until the next morning to get something fixed but I guess that's one of the consequences of these systems.

> If I know that a UX member is in a particular location I can get him to do a review by sending them a link to make it accessible […] and it's not the same quality of feedback, you have to plan around that.

(iv) With more frequent communication, "heavy" handovers of documentation are found to be reduced by providing more lightweight specifications that can easily be changed and don't take a lot of time to produce, thus creating a more of an efficient way of working.

> I have just completed a 'lean' spec document to supply to the developers for adding AV to the page. It took me considerably less time than the old way - I'm glad I've been pushing to work in this way. So as long as the developers are happy with it and we pair when they build this it should be a huge time saver going forward.

Theme Category 2.3: Co-creation of prototypes and in-browser style guides

(i) Working in close proximity together as pairs was perceived to increase the rate and the type of output of the pair. In particular, this included the production of prototypes and in-browser pattern libraries or style guides. This was found to be a positive change in contrast to the out of sync "heavy" handovers of design document from when the disciplines had worked in isolation.

> Getting into the browser and out of 'Sketch' quickly is super important, this really helps the process and especially the agile process because of the increase in efficiency and decrease in handover or documentation.

> [On working in browser together] So it's quite quick. It's good to get to quick solutions. Particularly with the motion designs.

> Use Pattern libraries etc. to increase Dev-Des communication and involvement.

> [About in-browser pattern libraries] A tool like this allows you to create things that are more directives and not as static as using something like illustrator which is what most designers only use.

> Making designs pixel perfect in his design file is just a waste of time because we've already got a lot of front-end all set up and ready to go.

(ii) The prototypes could be also tested or shared with the wider team and stakeholders to better inform their decision-making. The co-creation of prototypes in this way was perceived by the pairs as a big improvement to their process in contrast to the handovers of documentation or prototypes being created in insolation.

> This was the prototype that we made and then used to go and test with on Tuesday. We went Guerilla testing out and about.

> You have a prototype to test quickly with friends!

(iii) Pairing enabled co-designing in the browser to occur which was seen as a useful way of making decisions, seeing how different components would adapt and to iterate upon the UX.

Pairs could interrogate design or interaction features together e.g., on Sports results and data tables, weather information and audio / video player interactive elements. This helped the pairs to identify problems, including the breaking of elements in the front-end, the colour schemes and visual design of the work in the browser.

> We've built our codebase to be easy to develop on so it allows us to do that process easily early on. We've spent a couple of days just tweaking little things... and I don't see that as a bad thing I think it's a good thing because we're improving it and we're discussing issues that we would potentially miss otherwise, and we know it'll work because it's in the browser!

> Getting into the browser and out of sketch quickly is super important, this really helps the process and especially the agile process because of the increase in efficiency and decrease in handover or documentation.

Theme Category 2.4: Shared ideation, experimentation and iteration

(i) Early ideation and experimentation together helped to share ideas and define the increments of work. Initial idea generation around a problem was facilitated through the use of collaborative sketching and discussion before formalised design work took place.

Pairs A and B, who were located together started work by designing together and Pair A even had a joint sketchbook to work on ideas together.

> We actually have a joint sketch book that we both use. […] So... [shows sketch book] So this is just some ways [laughs]... so this is some initial ways of displaying tennis depending on a live event or different websites so we're just exploring the best way by sketching how to display information.

> On paper - couple of designers sketched ideas.

(ii) By experimenting and iterating together as a pair in a low fidelity and low-cost way early on, it helped to reduce bigger problems later on. Making work in the long run more efficient.

This practice counters the commonly talked about handover of heavily defined designed documents which might not entirely achievable.

Without iteration like this it would have gone too far down the pipeline and it would have seen to be broken in Test or worse when it was live.

Yeah, we want to get to a certain point when we can review it with other people like the creative director which is roughly every few days / weekly.

(iii) Despite Pairs C to E starting off seated separated from each other, they expressed the opinion that early informal discussion about features was useful for sharing ideas, identifying constraints, and generally helped to define the work aims and process. These aspects were even more improved when sitting physically close to one another.

Although we are teaching each other the skills that we have, we sit together and talk to each other about how things work, things that don't work and how to solve them.

So, then we'll quickly just go right let's just build it and see what happens. Because immediately he [the developer] will pick up on things that might be difficult to develop in the future, so you can address them straight away.

Theme Category 2.5: Paired work was seen to benefit whole process

(ii) The designers and developers in the study perceived that pairing together on projects was beneficial to their design and development processes.

It's been successful because we release new pages and designs without any negative feedback, it's been a good working relationship, and no one got upset or argued!

Quality of information and reliability of the service has improved – better offering for users!

It is a successful project mainly because we are working in pairs.

And when we are pairing you can see what is the best solution and they show you what they can do technically. […] So, it's quite quick. It's good to get to quick solutions.

We do it on the fly, pairing with [name], it works well.

(iii) It was found that designated pairing had a positive impact on the process because the discussions between the roles helped to analyse a design problem upfront as opposed to the previous method of designers handing over a design document without any discussion. This was perceived as to saving a lot of time and effort and increased satisfaction.

[On what pairing helps to prevent] We try to break that barrier down and not having this waterfall approach where designers bring a picture of a feature that they want and then the developers say, how are we going to do that, where does it even come from?

[On what would happen without pairing] …and then everyone goes into a big argument and in the end the designers, the developers don't get what they want, the product owner doesn't get want they want, editorial doesn't get want they want

I am looking at how to include playable AV content on the homepage...Today the developer I'm paired with picked up a small piece of work and basically built exactly

what I wanted based on conversations […] Although it was only a REALLY small piece of work it still feels like a bit of a breakthrough for the working process within the team.

(iv) It was remarked that pairing helped to overcome 'invisible' barriers between the two roles, allows the disciplines to rightfully crossover and overlap so that skills can be shared. This ultimately was felt to create a better product and thus a more successful, suitable UX.

Some Developers don't want to do front-end design and think UX is still outside the "team"! They are not 'they', they are 'us'!

We are separate disciplines, but we can all work together to make the product.

(v) It was perceived that learning skills about each other's role to understand what is possible is really useful. In particular, designers learning code is a positive driver in the team, to understand their developer and development process better and to improve communication with others about problems and solutions to help to deliver changes/functionalities quickly to users.

Designers find learning code useful to be able to talk more to Developers to respond rapidly to requirements changes. Saving effort when prototyping in code.

Technical reasons for Design bugs are identified early.

It's important to understand each other's constraints – as well as possibilities I would not have thought of!

Some Designers use it to create prototypes in browser […] But it's difficult for designer in practice to keep up and invest time in [once not paired anymore].

Theme Category 2.6: Joint decision-making

(i) Breaking down tasks to make small decisions and iterations together. In particular, pairing was beneficial when making "small" changes - especially when working at a component level. Iterations would occur that needed quick collaborative decision-making as the component was developed in the browser.

By collaborating together as a designated pair, it was felt that it was easier to make confident decisions together and in particular Pair D and E both acknowledged that if they were paired together more often, they would be able to make more informed decisions.

Partnering is how we call this sitting together, but it only seems to work with small changes.

Sometimes we'll build something or [name] might design something and then half an hour later we'll go for lunch or come back and then we'll discuss it and we might realise we want to do it completely differently... and we might try that and then the next day we'll come in and say actually we should do it another way.

(ii) The designers and developers mentioned that agile collaboration should consist of breaking down tasks, continuous improvement, team effort and collaboration.

Whenever this had occurred in the past it would result in valuable work through releasing partial features and reducing wasted effort.

> The key is to build small chunks at a time rather than creating a massive chunk of work that may no longer be relevant. For design is similar, you can work in the same way. So even if you can finish your design in Illustrator for example it is actually not finished because it has not been released.

> He [the designer] is right, and the code is cheap, and we can build it so quickly, let's not get bogged down in thinking oh this is going to take forever if it doesn't work let's just change it and throw it away and start again.

(iii) Wider decision-making was felt to be improved and more informed. The pairs would present their work together and share their responsibility in front of the stakeholders and their wider team. Their solutions would often be a working prototype and a solution that is robust and using live data, so it would be more realistic to make decisions about.

> Yeah, we want to get to a certain point when we can review it with other people like the creative director which is roughly every few days / weekly.

> We will build the way that we think will work and sometimes we'll build through the other ideas to show that they don't work, this is useful for the surrounding team to see that other options aren't viable.

> So today me and [the developer] presented our tennis sports data work to editorial. The meeting was really positive - generally they really liked what we had done and gave some useful feedback to help improve our work.

8.4.3 Theme Category 3: Further opportunities for better integration

This section describes where the pairs of designers and developers believed that there were further opportunities for better collaboration and improved integration between UCD and ASD.

Theme Category 3.1: Pairing helps to align development process to Agile spirit

(i) Overall using Agile methods were seen as a positive. It was seen as important for breaking down tasks into manageable components, continuous improvement and collaboration. In particular, ASD was perceived to be about releasing partial features to the audience and reducing waste so that it could be learnt about through the use of data and testing.

'Waterfall' methods were frustrating for the pairs and shown through the designers and developers not being in sync and often at different phases of a project. This resulted in a desire to be truer to the agile processes and it was mentioned that without "true" agile processes it indicated a lack of collaboration between the two roles.

> Agile is about being able to adapt, trying to do things in small amounts and iterating even if what we are building is not what stakeholders wanted, it is easier to implement changes, so we don't waste what we've done.

> But it [the organisation] doesn't really work in an agile way. If it was then pairing between UX and Dev would be stronger, and we would be working on smaller enhancements.

(ii) Using Kanban over Scrum was perceived to be more iterative and involved less "rituals" which suited the often quite ad-hoc nature of the pairs working together.

> In [the team] we get bogged down into rituals of scrum.

> The problem with scrum was that we were rushing to meet the deadline and we came up with lots of bugs and the next sprint would be all about fixing the bugs.

> [On scrum planning] Overrun often and take the full day […]. Wasting a day just planning.

(iii) The positive experience of Agile methods was tempered by the fact that teams often were in reality too big to be collaborative and to allow working in a truly agile or "lean" way. Consequently, there was a desire for this to improve, catering for a more agile way of working.

> I think Lean is more for small teams or a small product which is building up to something bigger, but when you already have a very big product, how do you work quickly when it takes so long to develop things. It's like a conflict of cultural feasible technologies, which makes things difficult.

Theme Category 3.2: Desire for more opportunities to sit in close proximity to enable collaboration

(i) The pairs that were not permanently located together had a desire to more frequently sit in close proximity with one another, aiding deeper collaboration between the roles.

> This could be that your sat directly next to a developer and officially put on a project together so then your practices would almost overlap, learning a little bit more about each other's practice.

(ii) Sitting together more frequently was desired by both roles to enable collaborative work that provides the ability to make small increments, changes and refinements to the product and UX.

> Looking at the homepage today I'm reminded at how many little inconstancies there across the page. I would love to be able to sit down with a developer and work closely with them to make some refinements, but product wouldn't allow this type of work as they are so focussed on large epics rather than small enhancement work.

> Had a good chat with a developer who was enthusiastic about me sitting with him tomorrow and refining some of the spacing across the page. Something I've been desperate to do but wasn't sure we'd get the chance to because of the way the work is organised. Hopefully sitting closer to the developers on a permanent basis from later in the week will help the overall quality of the page increase.

(iii) There was a desire from the designers and developers to sit together to reduce handovers and get comfortable at regularly collaborating side by side in the browser together to produce demos or prototypes.

> Just been talking to a developer and the senior designer on homepage and we have decided trailing 'designing in the browser' - UX aren't going to provide explicit and prescriptive spec documents for build, instead we are going to sit with the developers when they come to do the UX tasks. This will potentially save us time creating spec docs and potentially increase the quality of the homepage. Win win!

Theme Category 3.3: More pairing and multidisciplinary team work where roles can crossover

(i) All pairs expressed a desire to be physically working closely together. This was because of previous experiences in a multidisciplinary team where the benefits had been seen first-hand. Pair F did not acknowledge this at first, but once they had discussed their ideas during the Contextual Inquiry they expressed a desire to collaborate more on their project to further progress their ideas and to work together in producing prototypes.

> UX should not work in isolation from developers

> Okay, so what I might do depending on how much time you have, I was going to organise this session where we could look into these things and I wanted to have some kind of mock-up or prototype in the end which I was going to do, but now that I've seen that there is already something really useful here that you have done and that we could use... it would be really good if we could try this out.

> Working in pairs - issues when not sitting together.

(ii) Pair A and B both talked about sharing each other's responsibility for the UX and how they would frequently cross over the boundary of what is expected of their role. This togetherness would also be useful for when sharing with others and presenting their ideas to stakeholders as a "united front".

> Yeah this involved all the dev team and ux team... pretty much everyone... we all put in ideas and then all worked on them and converged to pick the best ideas... we then took those few ideas to our stakeholders to get their input and then we discussed them a bit more ourselves what was doable in like a day.

(iii) Pairs A and B talked about how being together as an entire product team was really helpful and it worked especially well when they employed methods such as weeklong design sprints because everybody was invested in the ideas that came out of the work and it was all based on product KPIs / objectives.

> We are separate disciplines, but we can all work together to produce the product.

> I wanted to work with him from the very beginning and build it together and use his design skills and my front-end skills to work collaboratively to create a better working flow and hopefully produce things quicker.

> But the beauty is that we have the entire department involved.

Theme Category 3.4: Using more data and insights to inform decisions

(i) There was a strong desire from both roles to use more data and audience / user research to inform their decision-making.

When data and audience research had been used it was thought to be beneficial by all of the pairs, but it was not as frequent as it should be. It was thought that the use of objectives (such as KPIs and OKRs) were a useful way of helping to guide decisions around what was created and released to the audience. These objectives could then be benchmarked and measured against to understand progress and improvements.

> We also get involved in testing. More of the user research testing which is good to get insights. Also, user testing after something has gone live so we can improve it. Useful to see how users interact.

> I want to see more feedback from the audience.

> We could do this really quickly with multivariate testing.

8.4.4 Summary of Pairing Study B findings

Theme Categories	**1. Team collaboration & integration challenges**	**2. Pairing experiences and closer collaboration**	**3. Further opportunities for better integration**
Pairing	*1.1 Perception of operational separation between the two roles*	*2. 1 Working in close proximity*	*3.1 Pairing helps to align development process to Agile spirit*
	1.2 Lack of direct communication between the roles	*2.2 Early and frequent communication*	*3.2 Desire more opportunities to sit in close proximity to enable further collaboration*
		2.3 Co-creation of prototypes and in-browser style guides	
Team	*1.3 A lack of shared understanding of each other's discipline*	*2.4 Shared ideation, experimentation and iteration*	*3.3 More pairing of designers and developers in multidisplinary team work where roles can crossover*
	1.4 Separation of location and a lack of proximity	*2.5 Paired work is seen to benefit whole process*	
Organisation	*1.5 Frustrations of a lack of shared decision-making*	*2.6 Joint decision-making*	*3.4 Using more data and insights to inform decisions*

Table 8—4: Subsumed themes from Contextual Inquiry & Diary Study

8.5 Discussion of Study 4

The current study focused on finding out about the collaborative relationship between pairs of designers and developers who were paired together for 6 weeks within the context of several ASD product teams across a large media organisation.

Findings from Pairing Study B confirms that success can be found when UX designers and software developers come together to work closely in pairs. However, the organisational structure and setup of the two roles and their surrounding teams is found to be a challenge for Agile collaboration and localised decision-making. The following sections discuss the main findings that are concerned with this alongside further discussion in Chapter 9.

8.5.1 Organisational separation is a challenge for close collaboration

The study found that one of the key challenges to regular pairing between UX designers and developers was the perceived organisational separation that was placed upon the team structure and roles. The pairs cited that there was a "wall between UX and Dev" where it was common for the roles to be working on different projects at the same time. This creates problems in terms of quality and efficiency and often would result in "more bugs going live". In particular, a designers' work would be finished before a developer would get involved. Developers found this frustrating, as the heavy handovers of "flat designs" would not suit the requirements or technical constraints.

The separation meant that communication was at a minimum and would only occur on a need-to-know basis, often in ceremonial Agile meetings (e.g., standups and sprint planning). It was typical that due to a lack of regular communication, designers would not understand some of the jargon and technical language that was being mentioned – further disconnecting the two roles. Electronic communication was used to compensate for the physical and structural separation between the roles but it was perceived to be cumbersome and inefficient. It was found that the pairs had learnt to try and avoid it due to previous problems that they had experienced.

The cited separation in teams and roles meant that working together was not ad-hoc enough and a lot of planning was required to do the work. Blame and tension between the two roles

would then arise when collaboration had not occurred and the delivery of the UX is not what the designers expected. The issue of separation is then compounded because the work priorities of designers are perceived to be different to those of developers.

8.5.2 Challenges in decision-making

It was found across the different teams that there were too many people involved in the contribution to the product, meetings and part of the decision-making process. The nature of a large organisation meant that communication and collaboration needed to occur across teams for different purposes, hampering the localised level of decision-making and autonomy of the pairs and their surrounding teams. Designers showed general frustrations about the lack of involvement from their stakeholders e.g. their creative director or product owner. There was an organisational requirement for them to "sign-off" and make key decisions about the work and this lack of involvement meant it slowed-down their progress. The developers had experienced frustrations about a lack of shared decision-making in the team too. It was perceived that a manager or person outside the immediate team took important decisions and the team structure would often change without any team discussion, causing confusion and a feeling of a lack of ownership. This extends findings from previous research (Drury-Grogan & O'Dwyer, 2013; Moe, Aurum and Dybå, 2012) on how decision-making processes affect Agile development processes.

Localised decision-making was also felt to be not interdisciplinary enough. So-called 'small' decisions (about relatively minor design issues) were found to often happen via Agile based collaborative 'review' methods or user testing methods in their disciplinary teams. For developers this would occur with group discussions such as Agile ceremonial 'reviews', stand ups and via electronic methods known as 'pull requests'. For designers, decisions would often occur via 'design critiques' in their design teams or via user testing methods such as usability lab testing or guerrilla testing. However, these methods were still restricted to each other's discipline and there was little evidence of crossover in this sort of decision-making.

Despite this, when examples of pairing were cited by the participants, small decisions were found to be taken immediately in pairs, "We both decided we were happy with it" - and new components can be shown quickly to stakeholders either with quickly made prototypes or

adaptations of the actual codebase to create style guides or pattern libraries. This decreases disagreements and discussions on many levels. There is less conflict over the user experience versus the quality of the code as both of the roles share an understanding of each other's processes and they share common goals. It appeared that by working in pairs it empowers local decision-making and the work of the pair holds more sway when trying to gain "sign-off" on bigger more challenging decisions from stakeholders because of the combined work of both disciplines.

Furthermore, the pairs strongly desired the ability to use more data and audience / user research to inform their overall decision-making. Both roles had previous positive experiences of using data and audience research to help inform decisions but it was not as frequent as they would like. It was thought that the use of objectives (such as KPIs and OKRs) were a useful way of helping to guide decisions in the wider team around what was created and released.

8.5.3 Co-location is key

Location was perceived by the pairs to be an important factor for successful collaboration. All pairs reported that being in separate locations was unsuitable to close collaboration (despite Agile routines and remote communication). Unfortunately this was found to often be out of their hands due to lack of space or the structure of their surrounding teams. When the roles were separated, even by just banks of desks it could detrimentally affect the working partnership between the pair. In particular, it was found that even moving to just another floor or another area of the same floor in the building disrupts frequency and the quality of teamwork.

Close proximity between UX designers and software developers provided more frequent communication, closer collaboration and increased ad-hoc knowledge sharing. By sitting either next to or in close proximity of each other, there was not the need to laboriously schedule meetings to discuss work as discussions could happen frequently alongside each other. It was found that communication tools (e.g, email, JIRA or instant messaging tools) simply could not compete with face-to-face conversation. Co-location was especially useful when discussing – and trying – design solutions as quick prototypes, such as trying out the

animation and interaction of elements and components in the browser or by sketching down ideas together.

Overall, sitting together more frequently was desired by both roles to enable collaborative work. However, in addition to the challenge presented by the team structure, the environmental setup where the roles work did not have the flexibility to suit the situational nature of collaboration between them and their colleagues. This makes it difficult to work in a space together, to share a desk or to sit side-by-side on a regular basis.

8.5.4 Adopting collaborative methods as a team

Overall, the pairs found using Agile methods beneficial. In particular this was about releasing partial features to their audience and reducing waste so that it could be learnt about through the use of data and testing. In the study, the waterfall-like examples of separation between the two roles, resulting in heavy handovers were found to be frustrating for the pairs. This caused desire to be more true to the Agile processes and increase regular collaboration between the two roles and the wider team. Using Kanban over Scrum was perceived to be more iterative and involved less "rituals" which suited the iterative nature of how the pairs worked together. The positive experience of Agile methods was inhibited by teams often being too big to be collaborative and work in truly Agile or "lean" way so there was a desire for this to improve, catering for a more Agile way of working.

The pairs talked about the benefits of spending time together as an entire multidisciplinary product team during projects. This had worked especially well when they employed methods such as 'design sprints' or 'sprint 0s' because everybody was invested in the ideas that came out of the work and it was all based on product KPIs and shared objectives. It was found that the disciplines in the team would then share each others responsibility for the UX and how they would frequently cross over the boundary of what is expected of their role. This togetherness would also be useful for when sharing with others and presenting their ideas to wider stakeholders as a "united front".

8.5.5 Benefits of paired work

Despite the challenge of the on-going organisational separation, the benefits of pairing in the study were clear to see with all of the pairs talking positively about examples of their pairing experience. By working in pairs, the designers and developers showed benefits through close communication and knowledge sharing that allowed them to discover problems early on in the design and development process. These problems could then be ironed out together by acting iteratively using quick in-browser solutions and creating prototypes to test their decisions with users or their wider team. The continuous ideation and iteration that was enabled through pairing was particularly apparent for responsive web components where design adjustments and refinements needed to be made in the browser as different components were built (e.g., the format of tables across differently sized devices). Pairs observed that by having regular side-by-side communication it had also helped to develop close relationships, permitting more effective collaboration together - "it feels like a family […] with lots of banter".

Another major benefit of pairing was the saving of time and effort because designers were no longer designing unachievable or unrealistic concepts or "mock ups" upfront. This meant that the developers needed less time to correct unspecified designs due to the regular back and forth interactions between the roles. Equally, the designer needed less time to work on the behaviour or interaction design of elements (e.g. responsive web components) as it could be tested immediately in the browser by working closely with the developer.

By breaking down the 'invisible' barriers and bridge the gap between the two roles it permitted the disciplines to crossover and overlap so that skills could be shared and joined up. Learning skills about the equivalent role to understand of what is possible was perceived as being really useful and important to the overall process. In particular, designers learning code was a positive driver in the team, helping them to understand their developer and the development process better and to improve communication with others about problems and solutions to help to deliver a better UX. Developers also wanted to have more input and feedback from users but most changes (e.g. to well established features) seemed to cause negative feedback from the audience so this caused ambivalence to how the process works.

8.6 Summary

In summary, as with the Studies 1, 2 and 3 this study continues in the vein of finding the integration of UCD and ASD in practice as a challenge to close collaboration. The organisational setting in which the UX designers and software developers work is structured into separate teams, where challenges arise through separation in location and through decision-making. Following on from Pairing Study A, this study involved 6 design and development pairs over the course of 6 weeks and find that success can occur between the roles and closer collaboration can be sought. With organisational support pairing between designers and developers hold perceived benefits that help to improve the overall integration between UCD and ASD alongside ultimately improving the product and UX.

8.6.1 Next steps

The next steps are to take the findings from this study along with the insights from the other empirical studies to further discuss the findings alongside the reported literature to reach defendable conclusions that answer the overarching research question(s) outlined in this thesis.

Chapter 9
Discussion

9.1 Introduction

Integrating UCD and ASD practices is a challenge. Previous work identified various reasons for this, such as the loose adherence to the related guidelines and principles, need for training and mentoring, and increasing management commitment or consultancy (see reviews by Brhel et al., 2015; Law & Lárusdóttir, 2015; Silva da Silva et al., 2012). The current results extend previous findings by pinpointing the co-location and close collaboration between two crucial roles: Designers and developers.

In this chapter the findings from the preliminary practice-led study, the literature review and the four research studies are discussed. The studies are discussed in terms of how they answer the main research question(s) and how they inform future collaboration practices between UX designers and software developers working in an organisational setting. The main area of focus for this thesis was developed through conducting preliminary practice-led research and a review of relevant HCI and SE literature. This established a key area for research that led to the main research question:

How do people integrate UCD with ASD in practice and what is the impact on collaboration between designers and developers in an organisational setting?

Opportunely, in conjunction with this as described in the Research Design (Chapter 4), the organisation of the researcher permitted that data collection could take place across the organisation to drive this research forward. So to answer the research question, a series of empirical studies of industry practitioner's opinions, processes, and determining factors surrounding collaboration took place. In-depth interviews were conducted with 9 stakeholders (Chapter 5) followed by an online survey with over 100 designers and developers (Chapter 6). The research was emergent in nature and the outcome of the studies helped to develop the subsequent research question(s) and investigations:

What does 'success' look like between pairs of designers and developers working in an UCD and ASD environment? What are barriers and what are consequences of successful paired work?

To answer these questions, Pairing Study A took place with a single design and development pair (Chapter 1) following by Pairing Study B, a broader more comprehensive pairing investigation with 6 design and development pairs (Chapter 8) from around the organisation.

This discussion combines the findings from the research studies to answer the research question(s) and to create a compelling argument that informs future collaboration practices and academic research between UX designers and software developers working in an organisational setting. This next section of the chapter discusses the UCD & ASD integration challenges found across the organisation (9.2). This is followed by a discussion of the perceived desire for closer collaboration and pairing (9.3). Then, in section 9.4 the factors for success between UX designers and software developers are detailed and discussed. The limitations of the research are then considered (9.5) before finally, the overall discussion is summarised (9.6).

9.2 UCD & ASD integration challenges

Across the studies, the findings show that the use of ASD practices is seen to be mostly positive, specifically releasing partial features and reducing waste by learning and iterating through the use of data and user testing. Despite these benefits, non-Agile, and more 'Waterfall' practices were still found to be commonly used across the teams. In particular, the separation between design and development teams was perceived to cause inefficiency and cumbersome documented handovers.

This promoted a desire to be more aligned with Agile processes and increase regular collaboration and pairing between the two roles and the wider team. The next sections discuss the main challenges to close collaborative work between designers and developers that was found across the studies.

9.2.1 Organisational structure and team culture

All of the empirical studies conducted demonstrate findings that the pre-defined organisational structure creates a separation and an "invisible" barrier between the UCD and ASD teams. This causes a perceived lack of close collaboration between the design and development roles and hinders the overall integration of the two approaches.

Before the four empirical studies were conducted and following a review the relevant literature, the preliminary practice-led study identified the organisational divide between the UCD and ASD team, where collaboration between designers and developers occurred mainly on a need-to-know basis. This would often result in a waterfall-like workflow with upfront design work and heavy "handovers" of design documentation, leading to slow processes and frustrations for both roles. Despite this being found, the study was preliminary in scope and only involved the researcher's immediate team so it was difficult to understand whether this separation in teams was common across the organisation. To investigate this further, Studies 1 – 4 provided empirical insights into the organisational setting and its effects upon collaboration between designers and developers.

From the stakeholders in Study 1, their perspective was that due to the organisation's structure of divisions, an "invisible" barrier to successful collaboration had developed because of too much separation between teams and disciplines. In particular, this was concerned with having separated UX design teams and ASD teams rather than completely embedded teams. The stakeholders felt that an organisational culture of "us" and "them" had developed between the two teams and was detrimentally affecting the collaborative practices and relationships between roles. In Study 2, the analysis of the survey responses shows that designers and developers both want improvements in how teams in the organisation are structured – in particular, the main concern for designers is improving wider teamwork and the sharing of information and knowledge, and for developers it is having access to designers, including a suitable environmental setup. Furthermore, in Pairing Study A and B, the designers and developers also thought that the current organisational structure created a significant barrier to collaboration between the two roles and hindered the working relationships between their teams.

The result of this separation, according to the stakeholders in Study 1, is that projects are at different stages, causing frustration and wasting time due to a lack of joined-up thinking and

knowledge sharing earlier on in the process. Also, the complex team structures would create the need for a lot of planning and invariably the release of new features to users is perceived to be slower than it should be. Additional planning in this case helps to find better integration between UCD and ASD and fits with previous reports of a need to frequently 're-align' work processes and product development plans (Brown et al., 2011). But in this case, the extra planning and re-alignment work slows the process down, causing apparent frustrations for the stakeholders. Despite this, stakeholders stated that a lot of the organisation uses iterative ASD methods (e.g., Kanban) to help in providing more flexibility to teams to speed up releases to the user but it has had mixed success, again due to the complex organisational structures at play.

The organisational separation is also evidenced through the thematic analysis of the survey comments and pairing studies and brings about three main effects:

(i) Firstly, the designers and developers complained about a lack of regular communication where they felt that UX work and software development was often not aligned and they did not feel part of the same team. This further supports the challenge of actively re-aligning with each other's work (Brown et al., 2011).

(ii) Secondly, the findings show that design work is perceived by both roles to be happening in too much of a plan-driven way (i.e. "upfront" before sprints) and is not iterative enough. This is highlighted because developers don't feel engaged and can't contribute early on in the process, causing frustration, as they would like to be more involved in the design and ideation of a project.

(iii) Another outcome of this, which reflects on what the preliminary study found, is that the large periods of 'up-front design' on projects would lead to the waterfall-like handovers of design documentation between roles, with an "over the wall" culture being mentioned regularly. This contradicts UCD and ASD principles of facilitating iterative design and development with high levels of collaboration and reduced documentation.

Overall, the findings support the view that the separation of the UCD team(s) from ASD team(s) is often a result of organisational culture and structure and results in suboptimal outcomes as described previously in Ferreira et al. (2011). This makes for an on going challenge for UX designers and software developers to regularly come together for

collaboration and knowledge sharing. Although this appears to be an undesirable effect of the organisational structure, there are arguments in the literature that support up-front design remaining separated from the ASD process. In particular, Chamberlain et al (2006) reports that conducting UX design ahead of development work helps to plan out and provide a comprehensive view of the system or service being created, helping to reduce problems later on (Ferreira et al., 2007c). Meszaros and Aston (2006, p. 6) also agree that "Emergent Design doesn't work very well for user interfaces." and propose that "Some Design Up Front seems to provide better guidance to the development team and provides earlier opportunities for feedback." The evidence from the research across the organisation found that the large amount of upfront design presents a challenge to collaboration – with a desire for closer collaboration methods that foster a more iterative approach to design and development.

Finally, there is a lack of recent studies across organisations of this size and complexity, and a potential limitation of this aspect of the research is that the results appear at first not to be readily applicable to other environments, which is often the nature of applied research. However, data was obtained from designers and developers and their stakeholders that worked over a wide variety of digital projects, and across a number of teams employing a variety of Agile styles and typologies. Furthermore, the findings reflect and add to findings from other studies on factors for successful ASD: providing opportunities for improved teamwork and collaboration (Chan & Thong, 2009); the importance of adequate environmental setup (Mishra, Mishra, & Ostrovska, 2012); and the role of organisational culture and management support (Chan & Thong, 2009; Jurca, Hellman, & Maurer, 2014).

9.2.2 Location and environmental setup

From the empirical studies conducted, in addition to the organisational separation of the UCD and ASD team(s), separation was also evident through a lack of physical location sharing. This was felt to be an important factor for successful collaboration and contributed to a number of challenges between the roles and the overall integration between UCD and ASD.

In Study 1, the in-depth interviews with stakeholders confirmed that the environment was found to be a challenge for ad-hoc collaboration between roles. Often space was at a premium and not a lot of "breakout" areas were available for group collaboration and discussions. This was perceived by the stakeholders to make an impact on being able to have regular communication between the roles.

In Study 2, the analysis of the online survey responses confirmed that developers were usually co-located in one designated team, whereas designers were often required to move between different teams and projects. This setup meant that planning, design and implementation requirements were not always easy to coordinate between the two roles. There was consequently a strong desire for more 'joined-up' thinking and discussions between the two roles. Qualitative analysis of the responses suggests that these factors for Agile satisfaction for both groups were associated with the level of pairing of roles and increased physical co-location.

Similarly, in both of the pairing studies, location was perceived to be an important factor for successful collaboration. All pairs reported that being in separate locations was unsuitable for close collaboration (despite Agile routines and remote communication). Unfortunately this was found to often be out of their hands due to the structure of their teams or a lack of space in the office. When the roles were seated in separate locations - even by just a few meters - it could detrimentally affect the working partnership between the roles. In particular, it was found that even moving to just another floor or another area of the same floor in the building significantly disrupts frequency and the quality of teamwork.

One of the most commonly found collaboration issues between designers and developers was the handover of design documentation (e.g. design visuals and mock ups) between the two roles following the up-front design work. Alongside the aforementioned team structure challenges, location was perceived to be a major factor in causing this issue for two main reasons:

(i) The regular separation of location resulted in designers and developers having fewer opportunities for discussions and ad-hoc communication. Their respective processes were often not aligned and there were large gaps between collaboration points so handovers were not very iterative or "lean".

(ii) A lack of shared location between the roles creates a barrier for regular awareness and involvement in projects, meaning that the knowledge of progress, decisions and the opportunity to intervene is reduced. In particular, at the beginning of a project, the roles complained about a lack of joined up thinking, missing an opportunity for knowledge to be shared.

Therefore, role separation seems to result in an "us" and "them" mentality, and related frustrations for both roles. This finding aligns with Lee et al. (2011), who identified that a challenge for integrating Agile and UCD in a distributed environment is to effectively support the sharing of documents and artefacts among team members who are physically separated.

Developers found frustration due to a lack of involvement where contingencies would not have been catered for in documentation and was often portrayed through "flat mock-ups". Time would be spent trying to un-pick questionable design decisions, creating a challenge during development cycles e.g., for complex responsive web components. Designers would be frustrated because the developers output would be quite different from their upfront design work. And although the designers and developers had come together during regular Agile ceremonies (e.g., stand-ups, planning meetings) and had used electronic communication tools (email, jira, messaging tools), it was perceived that it did not compensate for "face-to-face" communication in close proximity in the same location.

The current findings demonstrate that the location of designers and developers and their respective teams is a key factor in the integration of UCD and ASD, aligning with a recent systematic literature review by Salah et al (2014). The issues that have arisen from the studies across the organisational setting show many similarities in the literature. Fox et al. (2008) reported that in the case of non co-location the exchange of design got delayed, aligning with the frustrations of developers who felt that the up-front design work was too far removed from their everyday work. Sy and Miller (2008) report that physical separation introduced difficulties in communication (see also Albisetti, 2010), creating a lack of sense of team and generating trust issues with an "us" and "them" mentality.

Furthermore, these conclusions also correspond with Najafi and Toyoshiba (2008) findings that the geographical separation led to the exclusion of the UX designers from release planning, sprints and Scrum meetings. It led to a lack of knowledge of the implemented features in development cycles and for both roles a difficulty in understanding any overlapping opportunities and constraints (Najafi & Toyoshiba, 2008).

Despite of all the acknowledged issues presented by non-co-location, Lievesley and Yee (2006) refused to co-locate designers with the development team. This was due to a number

of reported issues, the need for designers at the initial iterations to employ extensive mental efforts to make sense of and synthesise diverse user interests, information and influences. In addition, this way designers could accomplish their work without the issues resulting from an unfamiliar and tension laden environment of the development team. However, in the studies conducted in this thesis, these findings contrast with the desires of both the stakeholders and the designers and developers who express the need for regular location sharing throughout the duration of a project. Lievesley and Yee (2006) also report that rigorous communication methods were employed to deal with the physical separation – further highlighting that location is a key factor in collaboration success between designers and developers.

9.2.3 Decision-making

In the studies conducted, decision-making processes constituted an important contributing factor in how designers and developers work, reflecting and adding to reports from other studies on factors for successful ASD and the crucial role of decision-making (Drury-Grogan & O'Dwyer, 2013).

In the in-depth interviews with stakeholders it was felt that the "sign-off" and risk-averse culture of the teams in the organisation was affecting decision-making and the motivation of individuals. The stakeholders felt this was down to lots of people being involved with many layers of decisions – also being made a challenge because different stakeholders have varying opinions. This was found to be frustrating and despite the approach not being very true to Agile methods of giving teams autonomy in decision-making, they felt a responsibility to deliver. As a complex organisation with a large and varied audience, the stakeholders stated that people were often averse to being truly Agile or lean in their approach to releasing features and then learning from their success or failure. Additionally, due to the organisational separation of the UX team(s) from ASD team(s), designers were reported by the stakeholders to often work alongside other UX teams in order to make decisions. This slowed down their work on the product experience and hampers progress with the development team.

The designers and developers confirmed that decision-making was a challenge in both the online survey and in the pairing studies. The online survey respondents perceived senior stakeholders' hierarchical decision-making process as a barrier to successful ASD. In Pairing Study B, the nature of a large organisation was thought to hamper decision-making and

autonomy of the roles at a localised level due to the requirements of working with multiple teams and dependencies. In addition, designers showed frustration about the lack of involvement from their stakeholders (e.g., their creative director or product owner), this was due to organisational requirement for them to sign-off and make key decisions about the work. Because of this lack of involvement from it slowed-down their progress. The developers had experienced frustrations about a lack of shared decision-making in the team too. It was perceived that a manager or person outside the immediate team took important decisions and the team structure would often change without any team discussion, causing confusion and a feeling of a lack of ownership.

The designers and developers also perceived that localised decision-making was not interdisciplinary enough. So-called "small decisions" (about relatively minor design issues) were found to often happen via Agile based collaborative 'review' methods or user testing methods in their disciplinary teams. For developers this would occur with group discussions such as Agile ceremonial 'reviews', stand ups and via electronic methods known as "pull requests". For designers, decisions would often occur via 'design critiques' in their design teams or via user testing methods such as usability lab testing or guerrilla testing. However, these methods were still restricted to each other's discipline and there was little evidence of crossover in this sort of decision-making. Co-location and pairing of designers and developers was again therefore seen as an opportunity to enhance more localised and autonomous decision-making.

Overall, the findings support the view that decision-making is a challenge at both a management level and at a localised level between designers and developers in the organisation. This causes frustration and the perceived "sign-off" culture, creating a risk-averse barrier to successful collaboration in the integration of UCD and ASD. Similarly, other research finds that barriers to successful ASD reside in a crucial component of the Agile philosophy: autonomy and localised decision-making. Drury-Grogan and O'Dwyer (2013) observed in their qualitative study (focussing on team meetings) that some team members influenced the decision-making due to their seniority or experience. Serrador and Pinto (2015) found that team experience (together with moderators such as quality of vision and complexity of projects) affected outcomes and stakeholder satisfaction. Nevertheless, decision-making processes in ASD remains a challenge which is not limited to this

organisation, with many other organisations either ignoring or lacking adequate decision-making processes (Highsmith, 2009).

9.3 Desire for closer collaboration and pairing

As discussed in the previous section of this discussion, the empirical studies find three significant challenges to the integration between UCD and ASD which impact upon successful collaboration between designers and developers; 1) the organisational structure, 2) the location and environmental setup and 3) the processes surrounding decision-making.

In the preliminary practice-led study, the researcher found that closer collaboration with the developer in the team brought about a number of benefits. In particular, the study found success through side-by-side communication, iterative decision-making and idea sharing early on in the project. The two roles adopted a paired-like approach where they worked so closely together that the lines between the two disciplines became blurred due to regular knowledge sharing and crossing over of skills. Despite the perceived improvements, the study was only preliminary in scope, featuring the immediate team of the researcher. Following the literature review, the in-depth interviews with stakeholders and the online survey between designers and developers, the findings demonstrated a significant desire to improve collaboration methods between designers and developers across the organisation to overcome the previously described challenges.

To be exact, the stakeholders felt that successful collaboration would be more frequent if designers and developers were embedded in the same team in the same location, if pairing was commonplace and if the two roles shared their knowledge frequently to make decisions supported by the use of evidence from data and research. The stakeholders had confidence in this vision based on previous experience of improved collaboration in their teams when people were given ownership of a project and trust from above. It was believed ownership could be improved by using more data and research to inform decision-making; including more regular usability testing, statistics, A/B or multi-variant testing methods, giving more confidence in decision-making as a team and thus reducing the perceived risk-averse culture in the organisation. It was thought that these methods could work in combination with shared team goals such as Key Performance Indicators (KPIs) or Objectives & Key Results (OKRs).

Additionally, the online survey results show that close collaboration and informal communication between designers and developers is also desired alongside co-location and pairing to enhance more localised and autonomous decision-making. Also, using the right tools between the roles alongside having an effective environment for collaboration was seen to be an important factor for close collaboration.

Furthermore, the literature reports on many studies (Dyba & Dingsøyr, 2008; Salah et al., 2014) that address possible barriers to a successful ASD implementation, it is significant that there are only few recent studies on how these two crucial roles interact and collaborate. Brown, Lindgaard and Biddle (2011) observed that much of the interaction time between these roles was used to "re-align" individual work progress to ensure a common understanding of the project aims and ensure product development plans were on track. Ferreira, Sharp and Robinson (2012) found in ethnographic studies that successful integration of Agile and UX work relies on attitudes and work practices such as mutual awareness, expectations about acceptable behaviour, negotiating progress and general engagement with each other. However, there is a lack of rigorous insight or evaluation (see a review by Jurca, Hellman, & Maurer, 2014) whether and how designers and developers differ in the reported attitudes and practices, and how their co-operation is determined in particular with regards to organisational structures and decision processes.

With this in mind the research question(s) were expanded to incorporate the subsequent investigations:

What does 'success' look like between pairs of designers and developers working in an UCD and ASD environment? What are barriers and what are consequences of successful paired work?

To inform these research questions, Pairing Study A and Pairing Study B were conducted with the aims of understanding the effects of pairing between UX designers and software developers and particularly what success could be found to inform future industry practices and find improved ways of integrating the UCD and ASD approaches. Importantly, the focus of the pairing studies was not about how the wider team acted or performed or how overall, the two approaches of UCD and ASD were integrated, but instead about the barriers and consequences of designers and developers collaborating together in an organisational setting.

9.4 Factors for success between designers and developers

In this section, the factors for success between designers and developer that have emerged from the analysis of Pairing Study A & B are extended and discussed. In particular, the findings show that successful collaboration between designers and developers can be found through the following six factors: 1) Close proximity, 2) Early and frequent communication, 3) Shared ideation and problem solving, 4) Crossing over of knowledge and skills, 5) Co-creation and prototyping and 6) Making joint decisions.

9.4.1 Close proximity is key

The pairing studies between designers and developers found location and close proximity to be a crucial factor for collaboration and is a significant facilitator for effective communication, regular ad-hoc discussions and reducing handovers between roles.

The seating arrangements and proximity varied between the pairs but they all found marked improvement in their collaboration practices thorough co-location, especially by sitting side-by-side. Even if the pairs were only on nearby desks, they found the proximity to be useful for ad-hoc discussions, awareness and stand-ups in comparison to the team separation they had experienced previously.

The benefits of co-location in integrating UCD and ASD has been previously reported by Salah et al (2014) as well as by Williams and Ferguson (2007) who observed that co-location simplifies collaboration and facilitates continuous communication, negotiation, knowledge sharing, and instant decision-making between designers and developers (Tzanidou & Ferreira, 2010). Not only is sharing location found to be appropriate during the short term in the studies but the pairs also desired to be co-located permanently. Fox et al. (2008) reported that in the case of co-location of UCD practitioners and developers the exchange of design is constant and on going. This is reflected in the study with the roles finding benefits from a "constant dialogue" between each other, made possible through close proximity of their seating. Similarly, co-location enables the designer to become more integrated with the ASD team permitting more joined-up thinking, discussions and iterations of the design and development of the UX (see also Hussain et al., 2009).

Overall, the studies show that location is a key factor in the integration of UCD and ASD, with success being found in the pairing studies through the co-location of UX designers and software developers. Furthermore, when co-location is not the case, it was not through lack of desire but rather, as Ferreira et al. (2011) also found, the distinct work group cultures and organisational policies that that appeared to reduce cooperation between the roles.

9.4.2 Early and frequent communication

Through the pairing studies, the findings show that direct communication between designers and developers early on in projects helps to reduce problems later on because of the building of relationships and the mutual understanding of their roles. This early and frequent communication enables the early sharing of ideas, problems and the challenges ahead in a project. The pairs perceive that a "constant dialogue" between each other through direct, face-to-face communication allows for frequent verbal exchanges that allow iteration on their ideas, ending up with better solutions.

As miscommunication is often a key factor, as Ozcen et al. (2010) and Park et al. (2008) discuss, where often designers struggle to communicate interaction. Pairing Study B shows that this can be mitigated somewhat by involving the developers early in the design process, as advocated by Salah et al. (2014). The pairs in the current studies were particularly positive about the effect of this early and frequent communication between each other, with them regarding the strong team relationships as a key outcome of the initial efforts of regular communication - "it feels like a family […] with lots of banter". The improved relationship would then help later on in projects at different stages and in particular, when more difficult discussions need to take place. These findings are assuring, as according to reports, on-going and continuous communication needs to be maintained between UCD practitioners and software developers in this way to avoid the occurrence of delays and bottle necks in the development process (Ferreira et al., 2007b).

Significantly, due to frequent, direct communication between pairs, there is less of a requirement for heavy documentation handovers. By having a regular dialogue, either side-by-side or in close proximity, over time it means that documentation is more lightweight or "lean" with less up-front design where the UX would be planned and discussed as a pair rather than passed between the roles. Kollman (2008) also found frequent interaction with

people reduces documentation and thus Agile team members perceive documentation as insufficient for communicating the interactive behaviour.

Outside of the pairings, it is also found that communication and discussions in multi-disciplinary groups through the use of Agile ceremonies (e.g. 'Stand-ups', 'Three Amigos' and 'Retros') remains really important in order to keep the wider team informed and on track. Additionally, when not in close proximity to each other, electronic communication was often used between the pairs to assign work between roles and individuals in the team but this was perceived to work due to the relationships that had already developed through direct communication. Some pairs, because of such frequent direction communication, would only assign each other work by collaborating and sharing out responsibility whilst sitting next to each other.

Overall, the closer, more frequent, early communication supports the integration of UCD and ASD as the UX designer and developer are constantly available, or "on call", to participate in discussions that are ad-hoc in nature, thus impacting upon both processes (McInerney & Maurer, 2005). Without this level of communication between the roles, it is found in the studies that an understanding of the overall vision and direction quickly breaks down – and is reported by Kollman (2008) as even being useless. Ungar and White (2008) re-iterate this point, that frequent communication of the design vision minimises rework and illuminates any integration issues early on in the process.

9.4.3 Shared ideation and problem solving

The pairing studies found that there are significant benefits in sharing ideation and problem solving between both disciplines during particular stages of a project.

Early ideation and experimentation together helps to share ideas and define the increments and iterations of work, especially within the confines of adding or improving features into an existing system or product. Initial idea generation around a problem was facilitated through the use of collaborative sketching and discussion before formalised design work took place. By experimenting and iterating together as a pair in a low fidelity and low cost way early on, it helps to reduce bigger problems later on. In the long run, making their work more efficient.

Pairs who had not collaborated in this way before found that the early informal discussion about features is useful for sharing ideas, identifying key constraints, and to generally help define the work aims and process. And as previously discussed, by being paired and sitting physically close to one another it was felt to further facilitate this. This practice of sharing ideation and problem solving is found to counter the previously perceived long upfront and heavy handovers of design work and helps to integrate the developers into the design process, furthering their understanding and input.

In many cases, designers would still need to spend time working on the overall 'design vision', either with other designers or other team members – this might occur during the reported 'Sprint 0' phases or via 'Design Sprints'. Previously this would often occur without the presence of developers, creating an "us" and "them" mentality and leading to the aforementioned handovers and frustrations between roles. However, by pairing, engaging and actively involving developers during initial ideation and problem solving – as also reported in the literature (McInerney & Maurer, 2005) – it helps to achieve better integration. Additionally, later on in the process, following more upfront design work, developers would feel invested in the work and would engage in subsequent usability testing and further design cycles. In some cases, developers would even take part in facilitating ideation sessions because they understood the UCD process. This "shared understanding of the design vision" is also emphasised in the literature, with Salah et al (2014) reporting that developers have to understand what they are expected to implement as soon as possible. In the addition to this, the current findings show that by involving developers in the ideation and problem solving phase of the UCD process they become more engaged and invested in the ideas, making successful integration between UCD and ASD more likely.

9.4.4 Crossover of knowledge and skills

The current research found that because of pairing and increased side-by-side communication, the sharing of their knowledge and skills was perceived to increase during the time the pairs spent together. This collective sharing of skills improved in turn their efficiency and quality of output of their work. For example, designers could gain a greater understanding of performance implications, which in turn informed their design work. Developers could get more involved with the design process and help to facilitate design workshops with wider stakeholders. Designers also work with the coding environment so that both roles can collaborate on the same codebase; driving more understanding and the creation

of front-end iterations and prototypes. Lastly, both roles confirmed that having training available in their counterpart's respective discipline had proved to be worthwhile.

The regular knowledge sharing opportunities achieved through pairing and co-location was perceived to be a significant for the pairs. As discussed, when the roles are too far removed from each other, due to either organisational separation or when design is worked on 'upfront' for too long, issues arise and a lack of knowledge sharing develops. In particular, the frustration is found through a lack of knowledge of technical limitations by the designers, contingencies are missed and the developer finds frustrations in not being aware of the design process. This results in heavy handovers of documentation and roles being out of sync with each other with little overlaps where the designers end up frustrated by the outcomes of the development work.

The higher level of knowledge sharing that is found to occur through pairing helps to break down 'invisible' barriers between roles and bridge separations, allowing the disciplines to crossover so that skills can be shared. The benefit of designers and developers picking up each other's skills in this way is confirmed by the literature. Moffett (2014) suggests including programming concepts as an integral part of a designer's training. Conversely, offering developers basic design training helps them fill in missing information by applying relevant design principles and Albisetti (2010) found that developers were more engaged when taking part in UI specifications. Moreover, an online survey conducted by Hussain et al. (2009) found that 75% of respondents believed developers can pick up HCI skills by pairing with a UCD professional, 66% mentioning that this can be achieved via training.

9.4.5 Co-creation and prototyping

A key and unexpected finding that was identified in Pairing Study A and B was the shared creation of prototypes between the design and development pairs. The participants emphasised the creation of the prototypes as a shared output of their work that they were both invested in. This contrasts with some of the literature that reports designers should be the ones to create prototypes and according to Chamberlain and Sharp (2006), they should be willing to "feed the developers" with prototypes.

The pairs both found a lot of value in working together to produce prototypes because it represents the design concepts in a realistic way, helping to improve their decision-making.

By producing the prototypes, they also had something that was closer to production if the concepts were taken forward. This level of prototyping was also partly driven from the improved knowledge sharing and more frequent communication that came with pairing. In particular, designers achieving basic coding levels are a positive driver in the team, to understand the development process better and to improve their ability to create realistic prototypes.

Furthermore, the analysis showed that the through close collaboration the pairs work together in the co-creation of other artefacts that include shared sketch books, wireframes and in-browser 'pattern libraries' and 'style guides'. These different aspects emphasise the high degree of collaboration between the pairs in producing shared artefacts that can then be used alongside the wider team for decision-making, user testing, or further discussions and iteration. In this context, Brown et al. (2011) show that designers and developers constantly perform "interactional alignment work" (Strauss, 1988) and that the collaboration process is "patterned around the use of artefacts" (Brown, Lindgaard, & Biddle, 2012), confirming the importance of producing artefacts, such as prototypes, together as a pair.

Further reflecting the findings in the pairing studies, other research also reports the benefits of prototyping early on in the process (Chamberlain et al., 2006; Coatta & Gosper, 2010; Meszaro & Aston, 2006; Ungar & White, 2008). By regularly producing prototypes early on in the process and having ready-to-use in-browser 'pattern libraries' means less effort is spent in producing documentation that would have previously been 'handed over' as part of up-front design work. Moreover, by creating prototypes early on in the development process, it gave other members of the wider team the opportunity to provide opinions and give feedback, allowing the pair to learn and iterate on their work. A working prototype using "real" data may carry more gravitas with their colleagues and increase the team's involvement. In particular, in terms of user testing, the pairs find benefits in using prototypes for regular 'guerilla testing' as it is easy to gather feedback quickly with users as opposed to arranging testing through lab testing.

Overall, the co-creation of prototypes is seen as a significant success factor in the close collaboration between designers and developers. It was found to be particularly important that the development of the prototype(s) was shared, contrasting with reports from Ungar (2008) and Sy (2007) who suggest that designers work one iteration ahead of developers

regarding prototyping. Despite this, both approaches could be applicable depending on the team and context but the significant finding shows that designers and developers perceive the co-creation of prototypes at the same time to be an effective method of collaboration. Similarly, supporting the report by Federoff et al. (2008) that suggests that designers should work in parallel with the development team. Interesting, for the co-creation of prototypes between designers and developers there appears to be a lack of reported examples of suitable environments or tools, which potentially leads to further research in this area.

9.4.6 Making joint decisions

The pairing studies found that success between designers and developers came through breaking down tasks to make "small" joint decisions and iterations together in combination with using data and audience research to inform bigger, more strategic decisions by team stakeholders.

As discussed, insights show that decision-making is an on going challenge, both as a managerial and a localised level in an organisational setting. Despite this, there appears to be a desire from both stakeholders and the designers and developers to improve this, providing more ownership to teams and reducing the risk-aversive culture found in the organisation. Equally, collaboration is core to an Agile team (Beck, Beedle, Bennekum, et al., 2001) Agile teams self-organise and are meant to contribute collaboratively to make decisions (Schwaber & Beedle, 2001) and (Nerur et al., 2005). The pairing studies show that significant success between designers and designers occurs by making decisions on "small" changes - especially when working at a component level. Iterations would occur that needed quick collaborative decision-making as the component (e.g. a responsive web component) was developed in the browser. By collaborating together as a designated pair it was felt that it was easier to make confident decisions together and in particular, as pairing experience is gained, it is acknowledged that more regular, more informed decisions can be made.

The studies also find that closer collaboration between the roles improves wider decision-making too. With pairs presenting their work as a "united front" and sharing their responsibility in front of the stakeholders and their wider team. Their solutions would often be a working prototype and a solution that is robust and using live data so it would be more realistic to make decisions about. For bigger, more risky, strategic decisions the studies show that, despite negativity shown with stakeholders often being the key decision makers, it was

found that they should in fact be made by stakeholders as it is their responsibility. This seems contradictory to working in an Agile way as Agile teams are purported to be flatter and more flexible, as Nerur et al. report (2005) but due to the nature of the organisation, with many layers of stakeholders and a large varied audience, the studies find that some decisions should involve more people beyond the ASD team to factor in all considerations.

To aid the decision-making process there was a strong desire from both roles to use more data and audience / user research to inform their decision-making. When data and audience research had been used it was thought to be beneficial by all of the pairs but it was not as frequent as it should be. It was thought that the use of objectives (such as KPIs and OKRs) were a useful way of helping to guide decisions around what was created and released to the audience. These objectives could then be benchmarked and measured against to understand progress and improvements, thus giving people the "confidence to make the right call".

9.5 Limitations

The research studies conducted focused on obtaining a broad insight and an understanding about UX designers and software developers and the determining factors that contribute towards successful collaboration in an Agile environment.

One limitation of this research was that the findings appear at first not to be readily applicable to other environments (or organisations), which is often the nature of applied research. However, data was obtained from designers, developers and their stakeholders (managers) over a wide variety of online content, and across a number of teams employing a variety of Agile styles and typologies.

Furthermore, the results reflect and add to findings from other studies on factors for successful ASD: the crucial role of decision-making (Drury-Grogan & O'Dwyer, 2013); providing opportunities for teamwork and collaboration (Chan & Thong, 2009); the importance of adequate environmental setup (Mishra, Mishra, & Ostrovska, 2012); and the role of organisational culture and management support (Chan & Thong, 2009; Jurca, Hellman, & Maurer, 2014).

Additionally, many other companies and organisations will likely find themselves in a similar situation as the observed environment in a crucial aspect: developers may benefit from closer collaboration with (usually outnumbered) designers (Ferreira et al., 2012). Similarly, risk-averse attitudes in large organisations are common, often entailing top-down control of project work that can derail successful ASD processes.

Another barrier in the working relationship between designers and developers – which is have not addressed here - may also lie in their different personalities. There are reports of differences in personality and style within software development teams (Capretz & Ahmed, 2010). Acuna, Gomez, & Juristo (2009) found that when student teams adopted Extreme Programming (XP) they decided on their own type of cooperation and they experienced the least conflicts and showed higher levels of job satisfaction. However, it is not clear whether software engineers are different from other groups. Beecham (2008) found in a review of 92 papers that just half of studies report that engineers are distinguishable from other occupational roles in terms of motivation. Vijayasarathy and Turk (2012) emphasise the importance of 'enabling factors' such as training and setting norms in the Agile environment are important for its success.

9.6 Summary

In what is found to be a complex organisational setting, there are a number of on going challenges that have an impact upon the integration of UCD and ASD. The views that emerge from the empirical studies conducted show that successful collaboration and integration between UCD and ASD is a challenge due to 1) the organisational structure placed upon teams, 2) the location and environmental setup and 3) the decision-making processes that are in place. Despite this, there is a strong desire for closer collaboration and informal communication between designers and developers alongside regular co-location and pairing to enhance more localised and autonomous decision-making.

The empirical evidence regarding the effects of closer collaboration and pairing suggest that successful collaboration between designers and developers can be found through facilitating the following six factors: 1) Close proximity, 2) Early and frequent communication, 3) Shared ideation and problem solving, 4) Crossing over of knowledge and skills, 5) Co-creation and prototyping and 6) Making joint decisions. This is limited by the constraints and

cultural factors of upheld by the complex organisation and further work to apply factors to more scenarios should take place.

Chapter 10
Conclusion

This thesis investigated the integration of UCD and ASD in a complex organisation to find out and determine how 'success' can be found through close collaboration and pairing between UX designers and software developers. In doing so it contributed to four empirical studies and the determining factors for successful Agile collaboration between designers of developers. In this chapter a final summary (10.1) of the research and the main contributions (10.2) is provided. In addition, suggestions for future work are highlighted (10.3).

10.1 Final Summary

The findings in this thesis answer the overarching research question: *How do people integrate UCD with ASD in practice and what is the impact on collaboration between designers and developers in an organisational setting*? It has been argued that better integration between UCD and ASD can be achieved through closer collaboration and pairing between UX designers and software developers.

This emerged from the findings of four empirical studies conducted in a large, complex UK based organisation. The motivation for this research followed a preliminary practice-led study and a review of the relevant literature. The researcher experienced, through practice, a series of challenges and opportunities in collaboration between designers and developers in their surrounding team. Furthermore, the literature showed that studies of the integration of UCD and ASD are heavily influenced by practice but few recent studies had investigated the settings and contributing factors of practitioners work. Motivated to understand more, two empirical studies took place to investigate industry practitioners' opinions, processes, and determining factors surrounding collaboration in an ASD environment.

The studies found that both groups, and their stakeholders, perceive that collaboration can be significantly improved through a higher degree of co-location and greater co-operation early on in projects. In particular, findings from the online survey suggest that satisfaction with Agile increases with paired collaboration of designers and developers, and encouraging teams

to work in iterative cycles. A novel finding was that these factors were not moderated by ASD knowledge or experience. Rather, satisfaction with ASD seems to be associated with the organisation of teams, collaboration and related decision processes. While developers emphasise the need for closer collaboration with designers, designers appear to perceive information flow in teamwork as the main factor. In addition, ASD is perceived as too much hampered by top-down decision-making and a persisting "sign off" tradition in the organisation.

In light of these findings and the perceived desire for closer collaboration and pairing between designers and developers, further research took place to answer the emergent research sub-questions: *What does 'success' look like between pairs of designers and developers working in an UCD and ASD environment? What are barriers and what are consequences of successful paired work?* Analyses of the pairing studies confirm what was illustrated in the first studies, but importantly show success factors for collaboration between the two roles when the organisational challenges are overcome.

In conclusion, this thesis argues that successful collaboration between designers and developers can be facilitated by focussing on the following factors: 1) Close proximity, 2) Early and frequent communication, 3) Shared ideation and problem solving, 4) Crossover of knowledge and skills, 5) Co-creation and prototyping and 6) Making joint decisions. These factors are crucially determined and empowered by the support from the organisational setting and teams where practitioners work. The key challenges to enable integration between UCD and ASD - and thus encouraging close collaboration between UX designers and software developers - are: 1) Organisational structure and team culture, 2) Location and environmental setup and 3) Decision-making.

These observations and insights extend finings from previous work (e.g., see reviews by Brhel et al., 2015; Law & Larusdottir, 2015; Silva da Silva et al., 2011) and have important implications for practitioners, organisations and further research in this area. While some details of the reported observations and results may be specific to the current sample, the diversity of teams - and the high number of participants – involved in the studies will likely extend the main findings to other complex organisations.

10.2 Contributions

The contributions of this research come from the preliminary practice-led research, a review of the current literature and analysis of the empirical studies, whose findings have been compared and synthesised in order to answer the research question(s). This work provides contributions that are described in this section.

To achieve closer collaboration between UX designers and software developers and thus improve the overall integration between UCD and ASD, large complex organisations should pay close attention to overcoming the following challenges that hinder collaboration between roles:

Organisation structure and team culture – The organisational structure of teams and divisions makes for an on going challenge for UX designers and software developers to regularly come together for collaboration and knowledge sharing. The findings from across the empirical studies illustrate a challenge with regards to the organisational culture and structure; it drives the separation of the UCD team(s) from ASD team(s).
In particular, this was concerned with having separated UX design teams and ASD teams rather than completely embedded teams. The stakeholders felt that an organisational culture of "us" and "them" had developed between the two teams and was detrimentally affecting the collaborative practices and relationships between roles. The evidence from the studies across this organisation found that the amount of upfront design is problem, involving heavy handovers between roles and that change is desired to find closer collaboration methods that foster a more iterative approach.

Location and environmental setup – Location and close proximity of designers and developers is found to be an important factor for successful collaboration and contributed to a number of challenges between the roles and the overall integration between UCD and ASD.

From the empirical studies conducted, alongside the divisional separation of the UCD and ASD team(s), separation was also evident through a lack of physical location sharing.
All pairs reported that being in separate locations was unsuitable for close collaboration. Unfortunately this was found to often be out of their hands due to the structure of their teams or a lack of space in the office. When the roles were separated, even by just banks of desks it could detrimentally affect the working partnership between the roles. In particular, it was

found that even moving to just another floor or another area of the same floor in the building disrupts frequency and the quality of teamwork.

Decision-making – In the organisation, decision-making creates challenge at both a management level and at a localised level for designers and developers. When many people and layers are involved in the decision-making process, frustration and inefficiencies between roles and teams increase.

Across the empirical studies, the findings illustrate that the perceived "sign-off" culture creates a risk-aversive barrier to successful collaboration in the integration of UCD and ASD and is an important contributing factor in how designers and developers work. The findings show that this was down to lots of people being involved with many layers of decisions. There is a desire for more autonomous decision-making amongst teams where the use of data and design research can better inform decisions around the use of shared goals (e.g., OKRs or KPIs).

By overcoming the important challenges that are presented by large complex organisations, the findings show that it enables better integration between UCD and ASD and thus encourages closer collaboration between UX designers and software developers. When designers and developers come together, collaborative success can be found by upholding the following factors:

Factors for successful Agile collaboration between UX designers and software developers in a complex organisation.

1. Close proximity is key - Designers and developers should work in close proximity with each other. They should find opportunities to sit side-by-side to enable close collaboration, frequent communication and reduce separation and handovers between the roles.

The findings from the studies show that location is a key factor in the integration of UCD and ASD, with success being found in the pairing studies through the co-location of UX designers and software developers, enabling side-by-side communication, regular ad-hoc discussions and reducing handovers between roles.

2. Early and frequent communication - Designers and developers should come together to communicate frequently with each other face-to-face, especially early on in a project. This helps to develop relationships, reduce handovers and enables the sharing of ideas, limitations and user insights.

Overall, it was found in the studies that closer, more frequent, early communication supports the integration of UCD and ASD. The pairs perceived that a "constant dialogue" between each other through direct, face-to-face communication allows for frequent verbal exchanges that allow iteration on their ideas and improved knowledge sharing, ending up with better solutions.

3. Shared ideation and problem solving - Designers and developers should come together to engage in ideation and problem solving. This helps designers to understand key constraints and for developers to become invested in the design vision and the needs of users.

The studies found that there are significant benefits in sharing ideation and problem solving between both disciplines during particular stages of a project. Early ideation and experimentation together helps to share ideas and define the increments and iterations of work, increasing engagement, overall understanding and reducing bigger problems later on.

4. Crossover of knowledge and skills - Designers and developers should spend time pairing together to increase their knowledge sharing and crossover of skills, helping to develop a shared understanding of their problems, challenges and opportunities.

The empirical studies found that pairs perceived that because of pairing and increased side-by-side communication, the sharing of their knowledge and skills was increased during the time they spent together. This collective sharing was valued because it reduced lengthy 'upfront' design work and it enabled the mixing of their skills to improve their overall process and output of their work.

5. Co-creation and prototyping - Designers and developers should co-create concepts, prototypes and in-browser component libraries to realistically represent and test design ideas, informing further iterations and wider decision-making.

The findings from the paring studies identified that the design and development pairs both found value in working together to produce prototypes because it represents the design concepts in a realistic way, helping to improve decision-making through discussion and testing. They also found value in co-creating other artefacts together including, sketches, wireframes and in-browser style guides and pattern libraries. Sharing the creation of these artefacts encompasses the knowledge of the designer and the developer to make the concept feel "real" to the wider team and users.

6. Making joint decisions - Designers and developers should break down tasks together to make iterative joint decisions and inform their work alongside using data and audience research to inform measurements and bigger, more strategic decisions by wider stakeholders.

Insights form the studies show that decision-making is an on going challenge, both as a managerial and a localised level in an organisational setting. Analysis of the pairing studies found that success between designers and developers comes via making regular "small" joint decisions alongside using data and audience research to inform bigger, more strategic decisions by team stakeholders.

10.2.1 The Paired Collaboration Manifesto

The empirical research studies conducted provide the main contribution of the research as described above, with the aim of informing better integration between UCD and ASD, improved collaboration between designers and developers and furthering the academic work and knowledge in the area.

The next section describes 'The Paired Collaboration Manifesto' – which leads on from the success factors outlined previously. This multi-applicable manifesto is targeted at industry practitioners as a way of disseminating the findings from this research for designers, developers and their organisational leaders within large dynamic Agile organisations and environments to improve practices and enhance the outcomes of collaborative work.

The Paired Collaboration Manifesto

Purpose

What we make is for everyone regardless of ability, location or the technology they use. So how can we make the experience the best it can be?
We believe the answer lies in happy, productive and collaborative teams. Teams where we harness our collective power to create the most innovative and relevant solutions.

Pairing beliefs

In our quest to nurture this culture of collaboration, these are the things we have come to believe. We believe in:

Regular pairing over disjointed workflows
Inclusive ideation over designing on an island
Reaching beyond your walls over going it alone
Sharing expertise over hoarding knowledge
Measured making & learning over releasing in the dark

Designers and developers make the perfect pair

True excellence is achieved together, through smart collaboration. That's why we strive to join up our thinking through regular pairing.

Ideation is not an island

Everyone has great ideas. That's why, from the off, we work as one team. Whether you're a designer, developer or something in between, everyone should feel empowered to contribute.

We solve problems side by side

Location should never be a barrier to communication. Share a desk, pick up the phone or hop on a virtual stand up. When we work without boundaries we work faster, better and closer together.

Shared responsibility is shared pride

There's a lot to be said for ownership when it comes to team motivation and morale. When everyone has a role to play, we can be proud of achieving great work together.

Mutual respect cultivates thoughtful solutions

By regularly sharing knowledge, expertise and methodologies, we can better understand each others' disciplines, improve our thinking and become excellent collaborators.

Understanding our canvas is key

We strive to develop a shared understanding of digital and its many platforms: their nature, qualities and capabilities. This means that together we can imagine and discover a broad range of opportunities for innovation.

Rapid prototyping creates faster solutions

We develop concepts quickly and rationalise them with prototypes instead of flat mockups. Working like this helps us refine our thinking, interrogate ideas and discover smart solutions more quickly.

Measuring success empowers decision-making

Design research and audience insight are a big part of what we do. By grounding our process in real data we can give everyone the confidence to make the right call.

10.3 Further work

Further work could extend the research in this thesis in the following ways:

10.3.1 Disseminate the factors for successful Agile collaboration between designers and developers

The research applied in this thesis and the contributions gained will be disseminated further to the wider academic community and industry; alongside the previously conducted talks, seminar and lectures (see Appendix D). This will include a journal paper and a magazine article to suit different audiences. By doing this it will help for other academics and industry professionals to apply and extend the research or create other approaches that change and push forward the way designers and developers work together.

10.3.2 Apply and extend factors for successful Agile collaboration between designers and developers

Extend this research to further studies with pairs of UX designers and software developers from across other contexts and circumstances. The specific factors or the Paired Collaboration Manifesto can be applied, tested and extended via experimentation and trials with UCD and ASD environments in industry. Additionally, the empirical factors for successful Agile collaboration between designers and developers can invoke further academic research studies. For example, it would interesting to quantitatively measure the direct input of pairing and collaboration on the output of designers and developers – possibly by taking two teams and comparing the overall UX at the end of a fixed period of time but with different factors involved.

10.3.3 Further investigate decision-making in UCD and ASD integration

Future research may aim to address how senior stakeholder's adoption of the Agile philosophy and the associated need to relinquish decision-making powers influences collaboration and cooperation in ASD teams. There is much evidence between the studies presented of the constraints of stakeholder management upon the effectiveness and perhaps the feasibility of collaborative pair programming within the organisation and perhaps represents a significant theme for further research. In particular, it would be interesting to

track where, how and when decisions are taken in the development process, and quantitatively measure their quality and outcome.

10.3.4 Investigate designer-developer personality traits and backgrounds

This study does not address the personality traits of designers and developers and their educational background as they enter the industry. What impact does their background have upon their role and how does it affect their collaborative practices with others? How could the educational system evolve to incorporate some of the key methods that designers and developers have exhibited in the empirical studies in this thesis? How can HCI/Design and SE/Computer Science students learn about working in a fast paced Agile environment and what is involved with working alongside different roles? Additionally, how might future education systems support the development of designers and developers to overlap their skills, bringing the roles even closer together as potentially a single discipline?

Overall this is an area that is rich in opportunities for further research. The designer-developer dynamic in the context of a fast paced agile environment provides a number of exciting future research opportunities. In particular, how close collaboration between pairs of designers and developers can impact upon the fields of HCI and SE around the world and how the development of the user experiences across new and evolving digital platforms can become more advanced and intuitive than ever before.

Appendix A

Ethics Documents

A.1 Letter of research consent

Below shows a copy of the consent that was granted by the organisation for the research to take place.

27 June 2016

Authorisation letter for data collected from research at the BBC (commissioned at the BBC) to be used as part of a University of Chester MPhil / PhD submission.

To whom it may concern

Project: Bridging the gap between Designers & Developers

Student: Alexander Jones (0816989)

Academic Supervisor(s): Professor Nick Avis & Dr John Kerins

The British Broadcasting Corporation ("BBC"), of Broadcasting House, Portland Place, London, W1A 1AA, is hereby giving permission for the data collected as part of an internal research study (known as "Noah's Ark") to be used as part of the submission for the above student research project.

This project has been independently internally reviewed by Gemma Newell, who has confirmed that the project is worthwhile and of scientific merit, and that necessary internal BBC ethical protocol has been followed.

Yours faithfully

Gemma Newell
Creative Director for Design Research

A.2 University Ethics Approval Documents

The document below is a copy of the University of Chester's ethical approval that was granted for the research undertaken for this thesis.

University of Chester

Faculty of Science and Engineering
Research Ethics Committee

Alexander Jones
30 Hewitt Street,
Chester,
Cheshire.
CH2 3JD

27th June 2016

Dear Alexander,

Study title: Bridging the gap between Designers and Developers within large agile organisations.
FSE-REC reference: 066/16/AJ/CS
Version number: 1

Thank you for sending your application to the Faculty of Science and Engineering Research Ethics Committee for review.

I am pleased to confirm ethical approval for the above research, provided that you comply with the conditions set out in the attached document, and adhere to the processes described in your application form and supporting documentation.

The final list of documents reviewed and approved by the Committee is as follows:

Document	Enclosed?	Appendix №	Version №	Date
REC application form	**Mandatory**			
List of references	**Mandatory**	Yes		
Summary C.V. for lead researcher	**Mandatory**	Yes		
Written permission(s) from the relevant organisations to undertake the research (eg. to use collected data for research purposes)	**Y ☒ / N ☐**	Yes	1	27/06/2016
Written confirmation that participants have agreed for their data to be used for the	**Y ☒ / N ☐**	4		

purpose(s) outlined in this proposal				
Other documents (Please specify below, as necessary)	Y ☒ / N ☐			
Example Participant Information Sheet	Y ☒ / N ☐	3		
Example Participant Consent Form	Y ☒ / N ☐	4		

Please note that this approval is given in accordance with the requirements of English law only. For research taking place wholly or partly within other jurisdictions (including Wales, Scotland and Northern Ireland), you should seek further advice from the Committee Chair / Secretary or the Research and Knowledge Transfer Office and may need additional approval from the appropriate agencies in the country (or countries) in which the research will take place.

With the Committee's best wishes for the success of this project.

Yours sincerely,

Eustace Johnson
Chair, Faculty of Science and Engineering Research Ethics Committee

Enclosures: Standard conditions of approval.

Cc. Supervisor/FSE-REC Representative

Appendix B

Collaboration Constraints for Designers and Developers in an Agile Environment

2.1 British HCI 2016 Conference Paper

Below is a copy of the British HCI 2016 Conference Paper - (see Jones et al., 2016)

Appendix C

Appendix to Pairing Study B

C.1 Overview of Contextual Inquiry Findings

The Contextual Inquiry was used to obtain information about the context of Design and Development practices and behaviours that were taking place at the beginning of the study. It was perceived by both roles that the team size was too big and the structure could often be difficult to understand, as changes would often occur unknowingly. This structural difficulty shows too from is a lack of input and transparency from stakeholders (line-managers) when decisions are required. From this it was apparent that Developers and Designers were birth frustrated by a lack of shared decision-making and team ownership.

Both roles felt that there was separation in collaboration between the two roles however when collaboration had occurred, value was felt by both disciplines in collaborating to break down tasks and releases. This lack of collaboration throughout the process showed by Developers expressing a desire to get more involved with the design process, whilst Designers found the speed of change and how much they can get involved with frustrating.

Developers also felt that the designer's contribution does not suit their requirements. Which could show that the two roles are not in sync with the other and knowledge is not being shared to help to understand each role or what each other are working on.

As the pairs began the study, the initial observations showed that all of the pairs except for one, sat separately. They all came across as very respectable of each others discipline and after a short time came together to work alongside each other, possibly because of the study guide. When the roles came together the benefits of pairing were clear to see straight away especially when making small changes together in the browser or by sketching out ideas and openly discussing it.

Whilst they were together they also attended several types of meetings which included a number of Agile stand ups, these meetings were uninspiring and often felt over complicated.

The Designers would often attend different stand ups to the Developers too. This contextual inquiry started the longitudinal study off by setting the scene and providing an understanding of how the designers and developers currently perceive collaboration with one another in their working environment.

C.1.1 Specific overview of findings of Pairs A – F

Pair A

Prototyping and building re-usable components in the browser together was part of their daily workflow. They frequently iterated and made localised decisions on the UX of the product together by sharing tasks between each other and combining their skills and knowledge. They often worked out an approach together in a low-fidelity way with frequent discussions and sketching ideas out together on a shared notepad.

They both have frustrations about some of the practices of the wider team and the practices of the other members of the team which they believe contributes to a general separation between UX and Dev and the decision-making and "sign-off" practices of their stakeholders. Despite this they have worked out that by producing in-browser prototypes and presenting their ideas as a "united front" it gives their ideas more credence with the stakeholders and the wider team. Side by side pairing is really important for the pair and the strong relationship that they have established over many months together is clear to see through their shared understanding and common beliefs in their collaboration methods. However, they do not possess strong collaborative relationships with their wider team which makes the pair acknowledge that they act like a single point of success or adversely a potential single point of failure if one of the pair is not present.

Pair B

Pair B work closely together frequently to create style guides and prototypes in the browser where they make quick iterations with the freedom to make small decisions about their work. Their team uses Kanban and occasionally adopts the Design Sprint method which is seen by the pair as a great way of involving and engaging the entire team at the beginning of a project. A lot of belief in communication being the key to success in collaboration and that working on a shared problem from the start e.g. via KPIs or OKRs, is equally important. Location is also recognised as an important factor for the pair but not necessarily by sitting side by side at all times but within the same vicinity. Collaboration and pairing together is

seen as being super useful but it happens in a more ad-hoc way than with Pair A. Their work will often involve checking in with other disciplines e.g. a Business Analyst, to sense check their progress and approach. They congregate around shared goals and the shared component library which is seen by both as a really important factor in their work.

Pair C

It is common for Pair C to be working separately on different aspects of a project and "over the wall" handovers of heavy specs are common. They would both like to be more joined up through regular pairing and creating components together to reduce the large amount of specifications that are being created – problems in of the handover process are particularly apparent and challenging with the advent of responsive web design as it requires many more versions of designs to be produced for different responsive "break points". During project work, they both found it difficult to find regular opportunities to get together and collaborative side-by side as their project work was often not at the same stage. The developer showed frustration by not being more involved in the design process. He also wanted communication to be more frequent up with less documentation being thrown "over the wall" and more shared component building from the beginning. When the roles do come together to collaborate or pair, they both note that it really helps the process and there is a lot of respect for each others role, the designer in particular benefits a lot from the knowledge sharing and the understanding of the development contingencies.

Pair D

Predominately the participants work in different locations (separated by two floors of the same building) and they both recognise that they would like to be more joined up and be part of the same team. They do come together to collaborate when needs must, however for the developer this often feels too little too late and can often ends up being like a sign-off meeting and he would like to be more involved in developing the ideas from the outset. They would both like this to be more of a regular activity through pairing and they believe that better collaboration will occur if everybody sits together, where the designers take turns to pair with the developers and the developers are included during the early stages of design work right through until delivery. This would create a more inclusive and shared process where every one can contribute.

Despite this, the UX designer is comfortable at producing prototypes as he believes it's a better way of producing more realistic designs rather than creating flat designs / mockups. The developer really likes this approach and it helps him to understand the designers vision with a lot more ease than unpicking specification documents. They both have a shared vision that it would be easier to collaborate around a shared pattern library in the browser which they believe will improve consistency and reduce documentation and prototyping which has become more difficult since the advent of Responsive Web Design, particularly across a large complex editorially based service.

Pair E

During the diary study Pair E show the most dramatic improvement in collaboration techniques with the adoption of pairing. They became a lot more joined up in their work and they both found success in closer collaboration between design and development. This was enabled by firstly changing their location so they were sitting closely to one another and this brought about more ad-hoc collaboration where they could make iterations in the browser together to help refine the UX. They both express that it was rewarding and would like to do it more often. By the end of the study they both have a strong belief that UX and dev should be collaborating from the early stages of a project by pairing regularly together to iterate and produce prototypes enabling knowledge sharing between the two disciplines. Despite this, the designer feels that the ability to roll out a pairing method like this regularly in his product would be challenging with the current stakeholder decision-making 'hierarchy'. The developer also expressed a desire for more training about the equivalents role e.g. to understand more about the design process.

Pair F

Pair F was very separated in their workflow and location but both have a strong desire to be more joined up and collaborate together more frequently but they work in separated teams within their product. This is largely due to a new piece of work being defined that leaves them focused on separate tasks. However, during the diary study they do find success in handing back and forth a prototype that is used for usability testing but they both recognised that they should have been working more collaboratively on this together to make the process more efficient. They both have a desire for faster, more focused collaboration between their roles but the opportunity to do so during the study was stipulated by the goals and timeframes of their product.

Appendix D

Achievements & publications

D.1 Presentations

2014, 2015 & 2016 – Lectures were delivered to Undergraduate students within the Computer Science department at the University of Chester. The lectures covered some of the broader and more fundamental elements of understanding the web the important practices of designers and developers working in industry.

February 2016 – BBC Develop Conference – I presented the outcomes of the empirical research studies at the annual BBC Developer conference in Salford in February 2016.

March 2017 – A lecture was delivered to Undergraduate students within the Computer Science department at the Lancaster University. The lecture covered some of the fundamental elements of understanding the web and the practices and techniques of designers and developers working in industry.

2015, 2016, 2017 – Postgraduate Poster Presentations – Took part in multiple Postgraduate Conferences at the University of Chester to present posters of the research at different stages. This provided useful academic feedback from research peers around the University.

D.2 Publications

June 2014 – Published a case study on the BBC Internet Blog of evolving the process between Design and Development for the 2014 FIFA World Cup.

June & July 2016 – The paper 'Collaboration Constraints for Designers and Developers in an Agile Environment' was accepted to the British HCI conference and was presented at the 2016 British HCI Conference in Bournemouth, UK.

Initially this was submitted as a segment of the research to the Computer Human Interaction Conference (CHI) and received peer reviews on the research. The paper was not accepted to the conference but the peer reviews helped to provide a steer and understanding of the validity of the research.

2015, 2016, 2017 – Presented the findings from the research and the 'Factors for successful Agile collaboration between UX designers and software developers in a complex organisation' to many teams and organisations, including: The BBC, the Guardian, Spotify, Code Computer Love and Real UX.

References

Ab, E. (2015). How Is the Software Development Process Impacted When a Large Company Goes Agile ?, 6–7.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: a comparative analysis. In *25th International Conference on Software Engineering, 2003. Proceedings.* (pp. 244–254). https://doi.org/10.1109/ICSE.2003.1201204

Acuña, S. T., Gómez, M., & Juristo, N. (2009). How do personality, team processes and task characteristics relate to job satisfaction and software quality? Information and Software Technology, 51(3), 627–639. Retrieved from http://dx.doi.org/10.1016/j.infsof.2008.08.006

Adams, A., Lunt, P., & Cairns, P. (2008). A qualititative approach to HCI research. *Research Methods for Human Computer Interaction*, 138–157. Retrieved from http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521870122&ss=toc

Adikari, S., McDonald, C., & Campbell, J. (2009). Little design up-front: A design science approach to integrating usability into agile requirements engineering. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5610 LNCS, pp. 549–558). https://doi.org/10.1007/978-3-642-02574-7_62

Aigner, W. (2009). Combining Extreme Programming and Interaction Design.

Alben, L. (2002). Quality of experience: defining the criteria for effective interaction design. *Interactions*. https://doi.org/10.1145/235008.235010

Albisetti, M. (2010). Launchpad's quest for a better and agile user interface. Lecture Notes in Business Information Processing, 48, LNBIP, 244–250. doi:10.1007/978-3-642-13054-0_26

Ambler, S. W. (2008). *Tailoring Usability into Agile Software Development Projects. Maturing Usability*. https://doi.org/10.1007/978-1-84628-941-5_4

Armitage, J. (2004). Are agile methods good for design? *Interactions*, *11*(1), 14. https://doi.org/10.1145/962342.962352

Bailey, P. H. (1997). Finding your way around qualitative methods in nursing research. *Journal of Advanced Nursing*, *25*(1), 18–22. https://doi.org/10.1046/j.1365-2648.1997.1997025018.x

Barbour, R. S., & Barbour, M. (2003). Evaluating and synthesizing qualitative research: The need to develop a distinctive approach. *Journal of Evaluation in Clinical Practice*. https://doi.org/10.1046/j.1365-2753.2003.00371.x

Beck, K. (1999). Extreme programming explained: Embrace change. Boston, MA: Addison-Wesley Longman Publishing.

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Manifesto for agile software development. Retrieved from http://agilemanifesto.org/

Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in software engineering: A systematic literature review. Information and Software Technology, 50(9–10), 860–878. Retrieved from https://doi.org/10.1016/j.infsof.2007.09.004

Belenguer, J., Parra, J., Torres, I., & Molina, P. J. (2003). HCI Designers and Engineers: It is possible to work together? *Software Engineering and Human-Computer Interaction*.

Bevan, N. (2008). Classifying and selecting UX and usability measures. In *Proceedings of the International Workshop on Meaningful Measures: Valid Useful User Experience Measurement (VUUM)*. https://doi.org/10.1086/663775

Beyer, H. (2010). User-centered agile methods. Synthesis Lectures on Human-Centered Informatics, 3(1), 1–71. doi:10.2200/ S00286ED1V01Y201002HCI010

Bias, R. G., & Mayhew, D. J. (2005). Cost-justifying usability: An update for the Internet age. San Francisco, CA: Morgan Kaufmann Publishers Inc

Bjørnson, F. O., & Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*. https://doi.org/10.1016/j.infsof.2008.03.006

Blomkvist, S. (2005). Towards a Model for Bridging Agile Development and User-Centered Design. In *Human-Centered Software Engineering Integrating Usability in the Software Development Lifecycle* (pp. 219–244). https://doi.org/10.1007/1-4020-4113-6_12

Bolger, N., Davis, A., & Rafaeli, E. (2002). Diary Methods: Capturing Life as it is Lived. *Annual Review of Psychology*. https://doi.org/10.1146/annurev.psych.54.101601.145030

Bono, J. E., Foldes, H. J., Vinson, G., & Muros, J. P. (2007). Workplace Emotions: The Role of Supervision and Leadership. *Journal of Applied Psychology*. https://doi.org/10.1037/0021-9010.92.5.1357

Borchers, J. O. (2001). A pattern approach to interaction design. *AI & Society*, *15*(4), 359–376. https://doi.org/10.1007/BF01206115

Boyce, C., & Neale, P. (2006). Conducting in-depth interviews: A Guide for designing and conducting in-depth interviews. *Evaluation*, *2*(May), 1–16. https://doi.org/10.1080/14616730210154225

Boyne, S. (2009). The phd literature review: its structure and contribution. *Proceedings of the Postgraduate Symposium 2009 Building Bridges in Social Science*, 229–315. Retrieved from http://www.academia.edu/413909/Boyne_S._2009_The_phd_literature_review_its_structure_and_contribution_in_Jolly_M._and_Wong_P._K._eds_The_Proceedings_of_the_Plymouth_Postgraduate_Symposium_2009_University_of_Plymouth_UK_pp_299-315

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77–101. doi:10.1191/ 1478088706qp063oa

Brewer, R. (2007). *Your PhD thesis*. Abergele, U.K. : Studymates.

Brhel, M., Meth, H., Maedche, A., & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. Information and Software Technology. Retrieved from https:// doi.org/10.1016/j.infsof.2015.01.004

Brown, D. D. (2013). Five Agile UX Myths. *Journal of Usability Studies*, *8*(3), 55–60.

Brown, J., Lindgaard, G., & Biddle, R. (2008). Stories, sketches, and lists: Developers and interaction designers interacting through artefacts. In *Proceedings - Agile 2008 Conference* (pp. 39–50). https://doi.org/10.1109/Agile.2008.54

Brown, J. M., Lindgaard, G., & Biddle, R. (2011). Collaborative events and shared artefacts: Agile interaction designers and developers working toward common aims. In *Proceedings - 2011 Agile Conference, Agile 2011* (pp. 87–96). https://doi.org/10.1109/AGILE.2011.45

Brown, J. M., Lindgaard, G., & Biddle, R. (2012). Joint implicit alignment work of interaction designers and software developers. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction Making Sense Through Design - NordiCHI '12* (p. 693). https://doi.org/10.1145/2399016.2399121

Bryman, A. (2004). Social Research Methods. In *Social Research* (Vol. 2nd, p. 592). https://doi.org/10.4135/9781849209939

Budwig, M. (2009). When User Experience Met Agile : A Case Study, 3075–3083.

Budwig, M., Jeong, S., & Kelkar, K. (2009). When user experience met agile. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09* (p. 3075). https://doi.org/10.1145/1520340.1520434

Butler, A. B., Grzywacz, J. G., Bass, B. L., & Linney, K. D. (2005). Extending the demands-control model: A daily diary study of job characteristics, work-family conflict and work-family facilitation. *Journal of Occupational and Organizational Psychology*. https://doi.org/10.1348/096317905X40097

Caballero, L., Moreno, A. M., & Seffah, A. (2016). How agile developers integrate user-centered design into their processes: A literature review. International Journal of Software Engineering and Knowledge Engineering, 26(08), 1175–1201. doi:10.1142/S0218194016500418

Cabage, N., & Zhang, S. (2013). Web 3.0 has begun. *Interactions*. https://doi.org/10.1145/2495249

Capretz, L. F., & Ahmed, F. 2010. Why do we need personality diversity in software engineering? SIGSOFT Software Engineering Notes 35(2), 1–11. Retrieved from https://doi.org/10.1145/1734103.1734111

Card, S. K., Moran, T. P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, *23*(7), 396–410. https://doi.org/10.1145/358886.358895

Cavaye, A. L. M. (1996). Case study research: a multi-faceted research approach for IS. *Information Systems Journal*, *6*(1 996), 227–242. https://doi.org/10.1111/j.1365-2575.1996.tb00015.x

Chamberlain, S., Sharp, H., & Maiden, N. (2006). Towards a framework for integrating agile development and user-centred design. *Lecture Notes in Computer Science*. Retrieved from http://www.springerlink.com/index/728680t2250vn864.pdf

Chan, F. K. Y., & Thong, J. Y. L. (2009). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, *46*(4), 803–814. https://doi.org/10.1016/j.dss.2008.11.009

Closs, S. J., & Cheater, F. M. (1999). Evidence for nursing practice: A clarification of the issues. *Journal of Advanced Nursing*, *30*(1), 10–17. https://doi.org/10.1046/j.1365-2648.1999.01043.x

Coatta, T., & Gosper, J. (2010). UX Design and Agile. *Queue*, *8*(11), 50. https://doi.org/10.1145/1874534.1891739

Constantine, L. L., & Lockwood, L. (2002a). Process Agility and Software Usability: Toward Lightweight Usage-Centered Design. *Information Age*, *8*(8), 1–10. https://doi.org/10.1.1.22.2229

Constantine, L. L., & Lockwood, L. A. D. (2002b). Usage-centered engineering for Web applications. *IEEE Distributed Systems Online*, *3*(3). https://doi.org/10.1109/52.991331

Constantine, L. L., & Windl, H. (2003). Usage-Centered Design: Scalability and Integration with Software Engineering. *Human-Computer Interaction: Theory and Practice*, *1*. Retrieved from http://www.foruse.com/articles/scalability.pdf

Cooper, A., Reimann, R., & Cronin, D. (2007). *About Face 3: The essentials of interaction design*. *Information Visualization* (Vol. 3). https://doi.org/10.1057/palgrave.ivs.9500066

Crawford, R. J., Garthwaite, P. H., & Howell, D. (2009). On comparing a single case with a control sample: An alternative perspective. Neuropsychologia, 47, 2690–2695. doi:10.1016/j. neuropsychologia.2009.04.011

Dahlem, M. (2014). Agile Practices in Practice - A Mapping Study -.

Dayton, D., & Barnum, C. (2009). The Impact of Agile on User-centered Design: Two Surveys Tell the Story. *Technical Communication*, *56*(3), 219–234.

Detweiler, M. (2007). Managing UCD Within Agile Projects. *Interactions*, *14*(3), 40–42. https://doi.org/10.1145/1242421.1242447

Draper, J. (2004). *The relationship between research question and research design. Crookes, Patrick A. and Davies, Sue eds Research into Practice: Essential Skills for Reading and Applying Research in Nursing and Health Care*.

Drury-Grogan, Meghann L. O'Dwyer, O. (2013). AN INVESTIGATION OF THE DECISION-MAKING PROCESS IN AGILE TEAMS. *International Journal of Information Technology & Decision Making*, *12*(06), 1097–1120. https://doi.org/10.1142/S0219622013400105

Dyba, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Ebner-Priemer, U. W., & Kubiak, T. (2007). Psychological and psychophysiological ambulatory monitoring: A review of hardware and software solutions. *European Journal of Psychological Assessment*. https://doi.org/10.1027/1015-5759.23.4.214

Edmonds, E. A. (1974). A Process for the Development of Software for Nontechnical Users as an Adaptive System. *General Systems*, *19*, 215–18.

Eerde, W., Totterdell, P., & Holman, D. (2005). Special section: Diary studies in work psychology. *Journal of Occupational and Organizational Psychology - J OCCUP ORGAN PSYCHOL*, *78*. https://doi.org/10.1348/096317905X40826

Erickson, T. (2000). Lingua Francas for design: sacred places and pattern languages. *3rd Conference on Designing Interactive Systems*, 357–368. https://doi.org/10.1145/347642.347794

Federoff, M., & Courage, C. (2009). Successful user experience in an agile enterprise environment. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5617 LNCS, pp. 233–242). https://doi.org/10.1007/978-3-642-02556-3_27

Federoff, M., Villamor, C., Miller, L., Patton, J., Rosenstein, A., Baxter, K., & Kelkar, K. (2008). Extreme usability: adapting research approaches for agile development. *Proceedings of ACM CHI 2008 Conference on Human Factors in Computing Systems*, *2*, 2269–2272. https://doi.org/10.1145/1358628.1358666

Ferre, X., & Moreno, A. M. (2004). Improving software engineering practice with HCI aspects. *Software Engineering Research and Application*, *3026*, 349–363.

Ferreira, J., Noble, J., & Biddle, R. (2007a). Agile development iterations and UI design. In *Proceedings - AGILE 2007* (pp. 50–58). https://doi.org/10.1109/AGILE.2007.8

Ferreira, J., Noble, J., & Biddle, R. (2007b). Interaction Designers on eXtreme Programming Teams: Two Case Studies from the Real World. *Proceedings of the Fifth New Zealand*, 1–8. Retrieved from http://www.greenstone.org/greenstone3/sites/nzdl/collect/nzcsrsc0/index/assoc/HASH1970.dir/doc.pdf

Ferreira, J., Noble, J., & Biddle, R. (2007c). Up-front interaction design in agile development. *Proceedings of the 8th International Conference on Agile Processes in Software Engineering and Extreme Programming*, 9–16. https://doi.org/10.1007/978-3-540-73101-6

Ferreira, J., Sharp, H., & Robinson, H. (2010). Values and assumptions shaping Agile development and User Experience design in practice. In *Lecture Notes in Business Information Processing* (Vol. 48 LNBIP, pp. 178–183). https://doi.org/10.1007/978-3-642-13054-0_15

Ferreira, J., Sharp, H., & Robinson, H. (2011). User experience design and agile development: Managing cooperation through articulation work. *Software - Practice and Experience*, *41*(9), 963–974. https://doi.org/10.1002/spe.1012

Ferreira, J., Sharp, H., & Robinson, H. (2012a). Agile development and user experience design integration as an ongoing achievement in practice. In *Proceedings - 2012 Agile Conference, Agile 2012* (pp. 11–20). https://doi.org/10.1109/Agile.2012.33

Ferreira, J., Sharp, H., & Robinson, H. (2012b). Agile Development and User Experience Design Integration as an Ongoing Achievement in Practice, 11–20. https://doi.org/10.1109/Agile.2012.33

Forlizzi, J., & Battarbee, K. (2004). Understanding Experience in Interactive Systems. *Human-Computer Interaction Institute*, *37*(2), 261–268. https://doi.org/10.1145/1013115.1013152

Fowler, M. (2005). The New Methodology. Retrieved from http://www.martinfowler.com/articles/newMethodology.html

Fox, D., Sillito, J., & Maurer, F. (2008). Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In *Proceedings - Agile 2008 Conference* (pp. 63–72). https://doi.org/10.1109/Agile.2008.78

Fox, R. (2012). Being responsive. *OCLC Systems & Services: International Digital Library Perspectives*, *28*(3), 119–125. https://doi.org/10.1108/10650751211262100

Frey, L. R., & Botan, C. H. (1999). Investigating Communication: An Introduction to Research Methods.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2002). *Design Patterns – Elements of Reusable Object-Oriented Software*. *A New Perspective on Object-Oriented Design*. https://doi.org/10.1093/carcin/bgs084

Garrett, J. J. (2011). *The Elements of User Experience : User-Centered Design for the Web and Beyond, Second Edition*. *Jesse James Garrett*. https://doi.org/10.1145/889692.889709

Gothelf, J., & Seiden, J. Lean UX: Applying lean principles to improve user experience. Sebastopol, CA: O'Reilly Media, Inc.

Gosper, J., Agathos, J., Rutter, R., & Coatta, T. (2011). UX Design and Agile: A Natural Fit? *Communications of the ACM*, *54*(1), 54. https://doi.org/10.1145/1866739.1866753

Gould, J. D., & Lewis, C. (1983). Designing for usability---key principles and what designers think. *CHI 83 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, *28*(3), 50–53. https://doi.org/10.1145/800045.801579

Grudin, J., & Fielding, R. T. (1995). Working Group on Design Methods and Processes. In *Proceedings of the Workshop on Software Engineering and Human-Computer Interaction* (pp. 4–8). London, UK, UK: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=645541.657989

Guion, L. a. (2006). Conducting an In-depth Interview 1. *Boards*, 1–4. https://doi.org/http://greenmedicine.ie/school/images/Library/Conducting%20An%20In%20Depth%20Interview.pdf

Hanington, B., & Martin, B. (2012). *Universal methods of design: 100 ways to research complex problems. Develop Innovative Ideas*. Retrieved from https://www.iitgn.ac.in/sites/default/files/library_files/2016/04072016.pdf%5Cnpapers3://publication/uuid/FD316216-3845-47B5-B7DA-0E68CD4B3014

Hassenzahl, M., & Tractinsky, N. (2006). User experience - a research agenda. *Behaviour & Information Technology*, *25*(2), 91–97. https://doi.org/10.1080/01449290500330331

Hektner, J. M., Schmidt, J. A., & Csikszentmihalyi, M. (2007). *Experience Sampling Method. Measuring the Quality of veryday Life. Journal of Chemical Information and Modeling*. https://doi.org/10.1017/CBO9781107415324.004

Henwood, K. L., & Pidgeon, N. F. (1992). Qualitative research and psychological theorizing. *British Journal of Psychology*, *83*(1), 97–111. https://doi.org/10.1111/j.2044-8295.1992.tb02426.x

Higgins, J. P. T., & Green, S. (2011). *Cochrane Handbook for Systematic Reviews of Interventions Version 5.1.0 [updated March 2011]. The Cochrane Collaboration* (Vol. 4).

Highsmith, J. (2009). Agile project management: Creating innovative products (2nd ed.). Addison-Wesley Professional.

Hodgetts, P. (2005). Experiences integrating sophisticated user experience design practices into Agile processes. In *Proceedings - AGILE Confernce 2005* (Vol. 2005, pp. 235–242). https://doi.org/10.1109/ADC.2005.24

Hussain, Z., Lechner, M. ., Milchrahm, H. ., Shahzad, S. ., Slany, W. ., Umgeher, M. ., … Wolkerstorfer, P. . (2012). Practical usability in XP software development processes. In *ACHI 2012 - 5th International Conference on Advances in Computer-Human Interactions* (pp. 208–217). Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84883224180&partnerID=40&md5=8b2080bd660e127fe946c4fb7a348393

Hussain, Z., Milchrahm, H., Shahzad, S., Slany, W., Tscheligi, M., & Wolkerstorfer, P. (2009). Integration of Extreme Programming and User-Centered Design: Lessons Learned. *Agile Processes in Software Engineering and Extreme Programming*, *31*, 174–179. https://doi.org/10.1007/978-3-642-01853-4

Hussain, Z., Slany, W., & Holzinger, A. (2009). Current State of Agile User-Centered Design: A Survey. *HCI and Usability for EInclusion*, *5889*, 416–427. https://doi.org/10.1007/978-3-642-10308-7_30

Iacono, J., Brown, A., & Holtham, C. (2009). Research Methods – a Case Example of Participant Observation. *The Electronic Journal of Business Research Methods*, *7*(1), 39–46. https://doi.org/10.1007/s10488-013-0528-y

Ilies, R., Schwind, K. M., & Heller, D. (2007). Employee well-being: A multilevel model linking work and nonwork domains. *European Journal of Work and Organizational Psychology*. https://doi.org/10.1080/13594320701363712

Illmensee, T., & Muff, A. (2009). 5 users every friday: A case study in applied research. In *Proceedings - 2009 Agile Conference, AGILE 2009* (pp. 404–409). https://doi.org/10.1109/AGILE.2009.45

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. Computers in Human Behavior, 51(0). https://doi.org/10.1016/j.chb.2014.10.046

Jones, A., Thoma, V., & Newell, G. (2016). Collaboration Constraints for Designers and Developers in an Agile Environment. In *Proceedings of the 30th International BCS*

Human Computer Interaction Conference: Fusion! (p. 37:1--37:9). Swindon, UK: BCS Learning & Development Ltd. https://doi.org/10.14236/ewic/HCI2016.37

Jurca, G., Hellmann, T. D., & Maurer, F. (2014a). Integrating agile and user-centered design: A systematic mapping and review of evaluation and validation studies of agile-UX. In 226 Proceedings - 2014 Agile Conference, AGILE 2014 (pp. 24–32). https://doi.org/10.1109/AGILE.2014.17

Jurca, G., Hellmann, T. D., & Maurer, F. (2014b). Integrating Agile and User-Centered Design A Systematic Mapping and Review of Evaluation and Validation Studies of Agile-UX. https://doi.org/10.1109/AGILE.2014.17

Kollmann, J. (2008). Designing the User Experience in an agile context. *Uclicuclacuk*. Retrieved from http://www.uclic.ucl.ac.uk/distinction-projects/2008-kollmann.pdf

Kollmann, J., Sharp, H., & Blandford, A. (2009). The importance of identity and vision to user experience designers on agile projects. In *Proceedings - 2009 Agile Conference, AGILE 2009* (pp. 11–18). https://doi.org/10.1109/AGILE.2009.58

Kruchten, P. (2001). Agility with the RUP. *Cutter IT Journal*, *14*(12), 27–33.

Landay, J. a., & Myers, B. a. (1995). Interactive sketching for the early stages of user interface design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '95*, (July), 43–50. https://doi.org/10.1145/223904.223910

Landsberger, H. A. (1958). *Hawthorne revisited: Management and the worker: its critics, and developments in human relations in industry*. Cornell University. Retrieved from https://books.google.co.uk/books?id=Obs5AAAAMAAJ

Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*. https://doi.org/10.1109/MC.2003.1204375

Latour, B. (1987). *Science in Action. How to Follow Scientists and Engineers Through Society* (Vol. 6). https://doi.org/10.1017/CBO9781107415324.004

Law, E. L., & Lárusdóttir, M. K. (2015). Whose Experience Do We Care About? Analysis of the Fitness of Scrum and Kanban to User Experience. *International Journal of Human-Computer Interaction*, *31*(9), 584–602. https://doi.org/10.1080/10447318.2015.1065693

Law, L.-C. (2003). Bridging the HCI-SE Gap: Historical and Epistemological Perspectives.

Lee, J. C. (2011). Evaluating eXtreme Scenario-based Design in a Distributed Agile Team, 863–877.

Lee, J. C., Judge, T. K., & McCrickard, D. S. (2011). Evaluating eXtreme scenario-based design in a distributed agile team. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11* (p. 863). https://doi.org/10.1145/1979742.1979681

Lee, J. C., & McCrickard, D. S. (2007). Towards extreme(ly) usable software: Exploring tensions between usability and agile software development. In *Proceedings - AGILE 2007* (pp. 59–70). https://doi.org/10.1109/AGILE.2007.63

Lee, J. C., McCrickard, D. S., & Stevens, K. T. (2009). Examining the foundations of agile usability with extreme scenario-based design. In *Proceedings - 2009 Agile Conference, AGILE 2009* (pp. 3–10). https://doi.org/10.1109/AGILE.2009.30

Leszek, A., & Courage, C. (2008). The doctor is "in" - Using the office hours concept to make limited resources most effective. In *Proceedings - Agile 2008 Conference* (pp. 196–201). https://doi.org/10.1109/Agile.2008.46

Lethbridge, T. C., Sim, S. E., & Singer, J. (2005). Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, *10*(3), 311–341. https://doi.org/10.1007/s10664-005-1290-x

Lievesley, M. A., & Yee, J. S. R. (2006). The role of the interaction designer in an agile software development process. In *CHI '06 extended abstracts on Human factors in computing systems - CHI EA '06* (p. 1025). https://doi.org/10.1145/1125451.1125647

Light, A. (2006). Adding method to meaning: A technique for exploring peoples' experience with technology. *Behaviour and Information Technology*, *25*(2), 175–187. https://doi.org/10.1080/01449290500331172

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*.

Löwgren, J. (1995). Applying design methodology to software development. *Proceedings of the 1st Conference on Designing …*, 87–95. https://doi.org/10.1145/225434.225444

Lurie, S. J., Schultz, S. H., & Lamanna, G. (2011). Assessing teamwork: A reliable five-question survey. Family Medicine, 43(10), 731–734. https://doi.org/22076717

Marcotte, E. (2010). *Responsive Web Design*. *Internetdocument*. https://doi.org/10.11635/2319-9954/1/1/18

Maudet, N., Leiva, G., Beaudouin-Lafon, M., & Mackay, W. E. W. (2017). Design Breakdowns: Designer-Developer Gaps in Representing and Interpreting Interactive Systems. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing - CSCW '17*.

McInerney, P., & Maurer, F. (2005). UCD in agile projects: dream team or odd couple? *Interactions*, *12*(6), 19–23. https://doi.org/10.1145/1096554.1096556

McMenamin, S. M., & Palmer, J. F. (1984). *Essential Systems Analysis*. Prentice Hall, Englewood Cliffs, NJ.

Memmel, T., Gundelsweiler, F., & Reiterer, H. (2007). Agile Human-Centered Software Engineering. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1* (pp. 167–175). https://doi.org/10.1177/004057368303900411

Memmel, T., Reiterer, H., & Holzinger, A. (2007). Agile Methods and Visual Specification in Software Development: A Chance to Ensure Universal Access. *4th International Conference on Universal Access in Human-Computer Interaction*, *4554*(2007), 453–462. https://doi.org/10.1007/978-3-540-73279-2_51

Meszaro, G., & Aston, J. (2006). Adding usability testing to an agile project. In *Proceedings - AGILE Conference, 2006* (Vol. 2006, pp. 289–294). https://doi.org/10.1109/AGILE.2006.5

Miller, L. (2005). Case study of customer input for a successful product. In *Proceedings - AGILE Confernce 2005* (Vol. 2005, pp. 225–234). https://doi.org/10.1109/ADC.2005.16

Miller, L., & Sy, D. (2009). Agile user experience SIG. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09* (p. 2751). https://doi.org/10.1145/1520340.1520398

Mishra, D., Mishra, A., & Ostrovska, S. (2012). Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation. *Information and Software Technology*, *54*(10), 1067–1078. https://doi.org/10.1016/j.infsof.2012.04.002

Moffett, J. (2014). *Bridging UX and Web Development: Better Results through Team Integration*. *Bridging UX and Web Development: Better Results through Team Integration*. https://doi.org/10.1016/C2013-0-13479-4

Moreno, A. M., & Yagüe, A. (2012). Agile user stories enriched with usability. In *Lecture Notes in Business Information Processing* (Vol. 111 LNBIP, pp. 168–176). https://doi.org/10.1007/978-3-642-30350-0_12

Myers, B. A. (1995). User interface software tools. *ACM Transactions on Computer-Human Interaction*, *2*(1), 64–103. https://doi.org/10.1145/200968.200971

Myers, M. D. (1999). Investigating Information Systems with Ethnographic Research. *Communications of AIS*, *2*, 1–19. https://doi.org/Need

Najafi, M., & Toyoshiba, L. (2008). Two Case Studies of User Experience Design and Agile Development. In *Agile 2008 Conference* (pp. 531–536). https://doi.org/10.1109/Agile.2008.67

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72–78. https://doi.org/10.1145/1060710.1060712

Newman, M. W., & Landay, J. A. (2000). Sitemaps, storyboards, and specifications. *Proceedings of the Conference on Designing Interactive Systems Processes, Practices, Methods, and Techniques - DIS '00*, 263–274. https://doi.org/10.1145/347642.347758

Nickerson, R. S. (1969). Man-computer interaction: A challenge for human factors research. *Ergonomics*, *12*(4), 501–517. https://doi.org/10.1080/00140136908931076

Norman, D., & Nielsen, J. (2016). The Definition of User Experience (UX). *Nielsen Norman Group Publication*. Retrieved from https://www.nngroup.com/articles/definition-user-experience/

Obendorf, H., & Finck, M. (2008). Scenario-based usability engineering techniques in agile development processes. In *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08* (p. 2159). https://doi.org/10.1145/1358628.1358649

Oettinger, A. G. (1967). The hardware-software complementarity. *Communications of the ACM*, *10*(10), 604–606. https://doi.org/10.1145/363717.363722

Ohly, S., Sonnentag, S., Niessen, C., & Zapf, D. (2010). Diary Studies in Organizational Research: An Introduction and Some Practical Recommendations. *Journal of Personnel Psychology*. https://doi.org/10.1027/1866-5888/a000009

Oppenheim, A. . (1994). Oppenheim, A. (1992). Questionnaire Design, Interviewing and Attitude Measurement. *Journal of Community & Applied Social Psychology*. https://doi.org/10.1002/casp.2450040506

Ozenc, F. K., Kim, M., Zimmerman, J., Oney, S., & Myers, B. (2010). How to support designers in getting hold of the immaterial material of software. *Chi*, 2513–2522. https://doi.org/10.1145/1753326.1753707

Palys, T. (2008). Purposive Sampling 1. *The SAGE Encyclopedia of Qualitative Research Methods*, *2*, 698–699. https://doi.org/10.1109/IPDPS.2011.210

Park, S. Y., Myers, B., & Ko, A. J. (2008). Designers' natural descriptions of interactive behaviors. In *Proceedings - 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2008* (pp. 185–188). https://doi.org/10.1109/VLHCC.2008.4639082

Patton, J. (2005). 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* (p. 266). https://doi.org/10.1145/1094855.1094961

Pinheiro, P., Silva, D., & Paton, N. W. (2000). UMLi: The Unified Modeling Language for Interactive Applications. In *International Conference on the Unified Modeling Language* (pp. 117–132).

Poppleton, S., Briner, R. B., & Kiefer, T. (2008). The roles of context and everyday experience in understanding work-non-work relationships: A qualitative diary study of white- and blue-collar workers. *Journal of Occupational and Organizational Psychology*. https://doi.org/10.1348/096317908X295182

Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction Design* (1st ed.). New York, NY, USA: John Wiley & Sons, Inc.

Raven, M. E., & Flanders, A. (1996). Using Contextual Inquiry to Learn About Your Audiences. *SIGDOC Asterisk J. Comput. Doc.*, *20*(1), 1–13. https://doi.org/10.1145/227614.227615

Reis, H. T., & Gable, S. L. (2000). (2000). Event-sampling and other methods for studying everyday experience. In. In *Handbook of research methods in social and personality psychology (pp*. https://doi.org/10.1016/j.jssc.2010.05.013

Rowley, J. (2014). Designing and using research questionnaires. *Management Research Review*. https://doi.org/10.1108/MRR-02-2013-0027

Royce, W. (1970). Managing the Development of Large Software Systems. *Proceedings, IEEE WESCON*, (August), 1–9.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modeling Language Reference Manual. The Unified Modeling Language Reference Manual*. https://doi.org/10.1017/CBO9781107415324.004

Salah, D., Paige, R. F., & Cairns, P. (2014). A Systematic Literature Review for Agile Development Processes and User Centred Design Integration. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14), London, United Kingdom, 13-14 May, 2014*, 5:1--5:10. https://doi.org/10.1145/2601248.2601276

Santosa, S., & Wigdor, D. (2013). A field study of multi-device workflows in distributed workspaces. https://doi.org/10.1145/2493432.2493476

Sapsford, R. (2007). What is Survey Research. In *Survey Research*. https://doi.org/http://dx.doi.org/10.4135/9780857024664.d5

Schwaber, K. (1997). SCRUM development process. In Business object design and implementation (pp. 117–134). doi:10.1007/978-1-4471- 0947-1_11

Schwaber, K., & Beedle, M. (2001). Agile software development with scrum (1st ed.). Upper Saddle River, NJ: Prentice Hall PTR.

Seffah, A., Forbrig, P., & Javahery, H. (2004). Multi-devices “multiple” user interfaces: Development models and research opportunities. *Journal of Systems and Software*. https://doi.org/10.1016/j.jss.2003.09.017

Serrador, P., & Pinto, J. K. (2015). Does Agile work? - A quantitative analysis of agile project success. *International Journal of Project Management*, *33*(5), 1040–1051. https://doi.org/10.1016/j.ijproman.2015.01.006

Sharp, H., & Robinson, H. (2004). An ethnographic study of XP practice. *Empirical Software Engineering*, *9*(4), 353–375. https://doi.org/10.1023/B:EMSE.0000039884.79385.54

Sharp, H., Robinson, H., & Segal, J. (2004). INTEGRATING USER-CENTRED DESIGN AND SOFTWARE ENGINEERING: A ROLE FOR EXTREME PROGRAMMING? In BCS- HCI Group’s 7th Educators Workshop: Effective Teaching and Training in HCI.

Silva da Silva, T., Selbach Silveira, M., Maurer, F., & Hellmann, T. (2012). User Experience Design and Agile Development: From Theory to Practice. *Journal of Software Engineering and Applications*, *05*(10), 743–751. https://doi.org/10.4236/jsea.2012.510087

Silva, T., Martin, A., Maurer, F., & Silveira, M. (2011). User-Centered Design and Agile Methods: A Systematic Review. *Agile Conference (AGILE), 2011*, 77–86. https://doi.org/10.1109/AGILE.2011.24

Singh, M. (2008). U-SCRUM: An agile methodology for promoting usability. In *Proceedings - Agile 2008 Conference* (pp. 555–560). https://doi.org/10.1109/Agile.2008.33

Sohaib, O., & Khan, K. (2010). Integrating Usability Engineering and Agile Software Development: A Literature Review. *Proceedings of International Conference on Computer Design and Applications (ICCDA)*, *2*(Iccda), 32–38. https://doi.org/10.1109/ICCDA.2010.5540916

Strauss, A. (1988). THE ARTICULATION OF PROJECT WORK: AN ORGANIZATIONAL PROCESS. *Sociological Quarterly*, *29*(2), 163–178. https://doi.org/10.1111/j.1533-8525.1988.tb01249.x

Sy, D. (2007). Adapting Usability Investigations for Agile User-Centered Design. *Journal of Usability Studies*, *2*(3), 112–132. Retrieved from http://www.upassoc.org/upa_publications/jus/2007may/agile-ucd.pdf%5Cnpapers2://publication/uuid/54FB82BC-3D3F-4B34-8E3C-7A02B631D618

Sy, D., & Miller, L. (2008). Optimizing agile user-centred design. In *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08* (p. 3897). https://doi.org/10.1145/1358628.1358951

Szalvay, V. (2004). An introduction to agile software development. *Danube Technologies*. Retrieved from http://www.danube.com/docs/Intro_to_Agile.pdf

Taylor, R. N., & Coutaz, J. (1995). *Software engineering and human-computer interaction : ICSE '94 Workshop on SE-HCI: Joint Research Issues, Sorrento, Italy, May 16-17, 1994 : proceedings*. *Proceedings of the Workshop on Software Engineering and Human-Computer Interaction*.

Thurmond, V. A. (2001). What Is Triangulation? The Point of Triangulation. Journal of Nursing Scholarship, 33(3), 253–258. Retrieved from https://pdfs.semanticscholar.org/ea21/2bd72920f4e57fde3b4aaf368f921021b1b2.pdf

Travis, D. (2011). ISO 13407 is dead. Long live ISO 9241-210! *Userfocus*, 1. Retrieved from http://www.userfocus.co.uk/articles/iso-13407-is-dead.html

Tschan, F., Rochat, S., & Zapf, D. (2005). It's not only clients: Studying emotion work with clients and co-workers with an event-sampling approach. *Journal of Occupational and Organizational Psychology*. https://doi.org/10.1348/096317905X39666

Tullis, T., & Albert, B. (2013). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics: Second Edition*. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics: Second Edition*. https://doi.org/10.1016/C2011-0-00016-9

Tzanidou, K., & Ferreira, J. (2010). Design and development in the "agile room": Trialing Scrum at a Digital Agency. In *Lecture Notes in Business Information Processing* (Vol. 48 LNBIP, pp. 372–378). https://doi.org/10.1007/978-3-642-13054-0_40

Ungar, J., & White, J. (2008). Agile User Centered Design: Enter the Design Studio - a Case Study. In *Extended Abstracts on Human Factors in Computing Systems (CHI '08)* (pp. 2167–2178). https://doi.org/10.1145/1358628.1358650

Vijayasarathy, L., & Turk, D. (2012). Drivers of agile software development use: Dialectic interplay between benefits and hindrances. Information and Software Technology, 54(2), 137–148. https://doi.org/10.1016/j.infsof.2011.08.003

Vinh, K. (2015). Design Tools Survey | The Tools Designers Are Using Today. Retrieved from http://tools.subtraction.com/index.html

Weiser, M. (2007). The computer for the 21 st century . *ACM SIGMOBILE Mobile Computing and Communications Review*. https://doi.org/10.1145/329124.329126

Wiemann, M. (2016). *Patterns as a tool for collaboration: A case study of collaboration between designers and developers through user interface pattern libraries*. Universitet Umeå.

Wilcox, E., Nusser, S., Schoudt, J., Cerruti, J., & Badenes, H. (2007). Agile development meets strategic design in the enterprise. In *XP'07 Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming* (pp. 208–212). Retrieved from http://dl.acm.org/citation.cfm?id=1768961.1769008

Williams, H., & Ferguson, A. (2007). The UCD perspective: Before and after agile. In *Proceedings - AGILE 2007* (pp. 285–290). https://doi.org/10.1109/AGILE.2007.61

Wixon, D., Holtzblatt, K., & Knox, S. (1990). Contextual design: an emergent view of system design. In *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90* (pp. 329–336). https://doi.org/10.1145/97243.97304

Wolkerstorfer, P., Tscheligi, M., Sefelin, R., Milchrahm, H., Hussain, Z., Lechner, M., & Shahzad, S. (2008). Probing an agile usability process. In *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08* (p. 2151). https://doi.org/10.1145/1358628.1358648

Yin, R. K. (2003). Case Study Research . Design and Methods. *SAGE Publications*. https://doi.org/10.1097/FCH.0b013e31822dda9e