Manuscript

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

Authors

Alexander Jones –University of Chester & BBC – alexander.jones@chester.ac.uk

Dr Volker Thoma – University of East London - v.thoma@uel.ac.uk

Keywords

Design; agile; software; development; collaboration; decision-making

ABSTRACT

Agile and User centred design processes have been reported to frequently putting contradictory demands on people working within these methodological frameworks. The current research addresses this point by focussing on the crucial relationship between software developer and designer. An online survey, a contextual inquiry and a diary study were employed with 107 developers and designers in a large media organisation to determine the factors for success in agile development cycles. The results from the survey show that while developers and designers have similar levels of satisfaction with agile processes, there are differences in the factors predicting those ratings. Developers are happier with the wider teamwork but want more access to and close collaboration with designers, while the latters' concern was the quality of the wider teamwork. Additional contextual inquiries and a diary study with pairs of designers and developers reflected the survey findings that close cooperation (and proximity) was essential for improving communication, reducing inefficiencies, and avoiding suboptimal products being released. However, organisational processes, the setup of the work-environment, and managerial traditions meant that this close collaboration and localised decision-making was found difficult to maintain. Results from the survey, the contextual inquiry, and the diary study found six factors for success from collaborations emerged.

1 INTRODUCTION

At the turn of the millennium two new approaches in information technology evolved into the fields of Human Computer Interaction (HCI) and Software Engineering (SE): User Experience Design (UX; or User Centred Design, UCD) and Agile Software Development (Agile) (for a brief review see Law & Lárusdóttir, 2015). The two approaches significantly changed the way teams designed and built software. 'Agile' is a summary term for a process in which the requirements (e.g., the users' or functional requirements) are addressed by applying solutions iteratively through collaboration between self-organizing, cross-functional teams (Dybå and Dingsøyr, 2008). 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 (see Salah, Paige and Cairns, 2014). These two popular approaches are often seen to put contradicting demands on teams and result in suboptimal processes and working conflicts. This paper focuses on collaboration and decision-making between pairs of UX designers (designers) and software developers (developers) as a potential means to address these differences between UCD and Agile methodologies. The current study took place in a large complex organisation in the United Kingdom (described further in Section 3).

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 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). Agile (Beck et al., 2001) emerged because of the need for a lightweight set of software development methods to address drawbacks of 'heavyweight' document-driven methodologies—such as so called 'waterfall' development frameworks (Szalvay, 2004).

UCD places the user and their requirements at the centre of the design process, aiming to involve them 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). 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 (Nielsen & Norman, 2015). 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).

Both UCD and Agile have their own specific sets of principles, practices and tools. Some of these, as described by Law & Lárusdóttir (2015) 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). The integration of these two approaches, which have co-existed for more than a decade, created a substantial amount of research interest. As described by Law & Lárusdóttir (2015) the application of UCD methods in software development processes has been very much encouraged and several methods such as Agile UCD (Beyer, 2010), Agile UX (Miller & Sy, 2009) and Lean UX (Gothelf & Seiden, 2013) are examples of such an effort to integrate the two approaches. Despite this, Agile methodologies and its variants (Dybå & Dingsøyr, 2008; see Inayat et al., 2015) are often perceived to be at odds with User-Centred Design (UCD) techniques (Salah, Paige and Cairns, 2014).

In a recent systematic literature review (Salah, et al., 2014) on Agile and UCD integration a number of main issues are identified that impact on design and development work. First, Agile 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 or any elaborate prototyping characteristic in UCD. Another issue is that the UCD work is not easily divided up in '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 modularization and iteratively adding features may be averse to their way of thinking and working.

Salah et al. (2014) also found in their review that the work-dynamic and relationship between designers and developers 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 & Dingsoyr, 2008; Salah et al., 2014) address possible barriers to a successful ASD implementation, there is a paucity of studies on how these two crucial roles interact and collaborate. This provided the initial 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?

In particular, this study aimed to explore 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. 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. Following the analysis of the online survey, the next research question was:

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 used where naturalistic insights were captured 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. By focussing specifically on designers and developers the study sought to gain an understanding of their context and environment, their communication tools and techniques, their methods for collaboration, the structure of their team, and any outside influences (e.g., stakeholders).

The article is structured as follows: In section 2, we present our literature reviews on UCD and ASD integration and any related literature on the designers and developers working relationship. In section 3 we describe our data collection and analysis methods. We then report our analysis and findings from our online survey in section 4. Stimulated by the results of the online survey, the design, implementation, and results of the Contextual inquiry and diary study are delineated in section 5. The empirical insights enabled us to identify determining factors for constraints and successful collaboration and between designers and developers; we discuss these factors in section 6. In section 7, we reflect on the limitations of our research work presented in this article and their implications for future work. Finally, the key findings of this study are then concluded in section 8.

2 RELATED WORK

In this section we present concise literature reviews of the related work on the integration of UCD and ASD and any previous findings into the relationship and determining factors for successful collaboration between designers and developers.

2.1 Background

The Agile Software Development (ASD) and User Experience (UX) processes are now commonplace within industry, especially in large multidisciplinary teams. And whilst there has been research that has looked at how UX and Agile combine (Budwig, 2009), there is very little research about the specific collaboration methodologies between people working in a UX design role and the people working in a software engineering role. Furthermore, there appears to be a clear gap in knowledge of what successful collaboration looks like between designers and developers and specifically what practices contribute towards a truly successful online experience.

2.2 Similarities of UCD and ASD

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

2.2.1 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, McCrickard, & Stevens, 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.

2.2.2 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, Robinson, & Segal, 2004).

2.2.3 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 (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 'Extreme Programming' (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 (Hussain, Milchrahm, et al., 2009), whereas in ASD, the focus is automated testing which can be time consuming and difficult to implement (Constantine & Lockwood, 2002). Furthermore, 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.

2.3 Integration challenges of UCD and Agile

This section compares some of the principles of UCD with those of Agile to understand where there is a challenge in integrating the two processes.

As discussed, UCD & ASD processes are now commonplace within industry, especially in large multidisciplinary teams. The UCD principle of *designing for the whole user experience*, is not compatible with Agile from the outset - in the sense of with Agile, work is divided up into chunks and smaller releases, making it difficult to consider the holistic design at all times. Furthermore, in Agile is it common to have tight deadlines (e.g. a two-week sprint) so the ability to keep users involved in the evaluation of the product or experience as it is designed iteratively becomes more challenging and ends up being cutting short.

In Agile, one of the most common methods is Scrum, which has been regarded as a process that includes UCD practices because of its emphasis on user involvement through user stories, and by its iterative and communicative nature (Schwaber, 1997). However, the UX component in Scrum is not an obligatory part of the process, which means that user involvement and evaluation is not guaranteed (Da Silva et al., 2012; Salah et al., 2014). Agile teams often commonly lack UCD representation because of a UX staff shortage (e.g., UX specialists from a centralised UX department have to work for several teams simultaneously). Furthermore, to gain an explicit understanding of users, tasks and environments – UCD work requires insights into potential users of a system or service over time to make sense of their varying emotions, abilities, contexts, etc. To do this medium to long term ethnographical studies are prescribed which may clash with the fast-paced release driven nature that Agile and Lean approaches uphold. Without aiming to provide an exhaustive list, Table 1 summarises some of the main challenges of ASD and UCD integration as identified by some major recent review articles in this area (Caballero et al., 2016; Bhrel et al., 2015; Da Silva et al., 2012; Law & Larusdottir, 2015; Salah et al., 2014).

TABLE 1
UCD and ASD integration challenges & conflicts and suggested solutions (implications for designers and developers)

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, 2013; 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., 2013 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., 2013)		
Lack of documentation	Decisions based on information; data, reports, prototypes	Artifact-based; web/wiki-based documentation; frequent interaction
(see Salah et al., 2013 and Kollmann, 2008)		
Limitation of work-in-progress	Delays UX designer from giving effective feedback on design	None
(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 Bhrel, 2015)		
Using tools/default metrics to measure work progress	UX professionals cannot easily track the interplay between user evaluation and redesign	None
(see Law & Larusdottir, 2015)		

2.4 The collaboration between Designers and Developers

As discussed, while there are many studies (Dybå & Dingsøyr, 2008; Salah et al., 2014) that address possible barriers to a successful ASD implementation, there are only few studies on how the two crucial roles of the designer and developer interact and collaborate together. 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. But there is still 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 organizational structures and decision processes.

This study therefore aims to investigate UX designers and software developers who are already supposed to be working using UCD and ASD approaches. The aim is to learn how their roles differ in the ASD process and how they perceive its outcomes.

3 RESEARCH DESIGN

This section describes the context, the participants, the data collection and finally the analysis methods that were used for each study. More specific details for each study are described in sections 4 and 5 and the following describes the main events for each and how each was informed by the findings from the former analysis.

The research took place within the context of a large, complex media organisation based in the United Kingdom with 18,000 employees. It includes several very large online products with an output of editorially steered online 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. 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.

To select participants 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. 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 have, by design, adopted the ASD process to varying degrees of flexibility; some involving mixed approaches to Scrum and Kanban methods.

To answer the research questions outlined in the introduction, a mixed method research approach was selected. This resulted in different data collection methods that were used for the two main studies. The methodology and type of data collected are outlined in Table 2 below.

TABLE 2
Data collected for each study

	Study 1: Online survey	Study 2: Pairing study
Method	Online questionnaire	Contextual Inquiry and Diary study
Data Collected	Quantitative (including ratings) and qualitative data (open responses)	Transcriptions, field notes and diary entries

As shown above, the data was captured using both qualitative and quantitative methods. For the quantitative analysis aspect of the online survey, a multivariate approach (of variance analysis, MANOVA) was applied to the rating scores of the survey covering the 3 main sections of the questionnaire – about satisfaction, teamwork, and collaboration, with the role of participants (designers and developers) as independent factor.

The qualitative data from the sources 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 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. These themes can then be discussed as findings in context of the other research and research aims.

4 STUDY 1: ONLINE SURVEY

In Study 1, we asked designers and developers 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 (variable ASD satisfaction), how well the wider team (including other roles, e.g. business analyst) worked together (variable teamwork), and whether there were specific issues in designer- developer collaboration and communication (variable collaboration) that would impact on the current processes, their goals or objectives. To do this we created an online survey following some initial qualitative interviews with stakeholders from the various teams.

4.1 Methods

As discussed in section 3, the study took place across a large media and broadcasting organisation based in the United Kingdom, with ca. 18,000 employees over all.

4.1.1 Participants and Procedure

There were 109 participants, (24 women) in the sample. We asked for a broad job role description to classify them as designers (n=54) or developers (55). We did not ask 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'. To select participants for the questionnaire we chose a sampling method known as 'purposive sampling' (Palys, 2008). This enabled us to focus our study 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.

All of the product teams that took part in the study claimed to have adopted the agile process to varying degrees of flexibility; some involving mixed approaches to Scrum and Kanban methods. Before we started data collection with designers and developers we interviewed nine stakeholders from the organisation (and one external expert on ASD). Together with the background research from the literature, we used the results of the interviews to derive a set of questions for designers and developers that would tap into commonly observed issues in agile collaboration. The interview data from the senior stakeholders are not reported here.

4.1.2 Measures

We employed rating scales to pre-defined questions as well as open questions (derived from the stakeholder interviews) with the goal to gain a broad insight from across the organisation and its teams and disciplines. Data was gathered using an online questionnaire tool (Typeform - http://www.typeform.com) that provided user-friendly participation across multiple devices from a single URL. The Likert scales used a 7-point scale ranging from "Strongly Disagree" to "Strongly Agree" with the middle option of "Neither agree or disagree".

The first set of questions were standard biographic questions, including the length of employment (in months) and a question on their perceived level of ASD knowledge. There were then 5 items to rate on the perception of quality of ASD processes and outcomes across the organisation and teams: the degree to which they agreed that ASD was working in their team, in the whole organisation, in their own product area, how well it served purposes such as achieving successful Responsive Web Design solutions, and how strongly they agree (or disagree) that ASD could be improved. Then followed four validated questions on teamwork which were taken from Lurie, Schultz, and Lamanna (2011). Participants were asked to what degree they were 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 agreed or disagreed that they felt able to act on a team vision (a 5th question item on leadership from the original scale was not used). This was followed by 5 questions specifically on the working relationship and collaboration between designers and developers. The questions – derived from the interviews with senior stakeholders and 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. In addition to being asked to rate these questions between 1 and 7 each of the survey sections could be commented on in a free form entry field as well.

4.2 Results

4.2.1 Quantitative analysis

Of the 109 responses, 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 (see Table 3).

TABLE 3
Participant's length of employment (years) and self-rated (1 to 7 scale) familiarity with ASD in each role

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

Responses were transferred into Excel and SPSS for further analysis. Raw data were screened for missing values and possible outlier influences, and normal distribution, and equality of variances. First, length of service (in days) and familiarity with ASD concepts were analysed for group differences. There was no difference in length of service, t (105) <1, with means of 814 days (SD = 852) for developers and 577 days (SD = 844) for designers. However, there was a significant difference in the amount of perceived knowledge about ASD concepts. Developers scored significantly higher, t (105) = 4.02, p < .001 (M = 6.5, SD = 0.57) than designers (M = 5.8, SD = 1.23); Mann–Whitney U = 947.5, p < 0.001, two-tailed).

The main quantitative analysis was applied to the rating scores (raw scores ranging from 1 to 7). This covered the 3 main sections of the questionnaire – satisfaction, teamwork, and collaboration. These ratings were analysed using a multivariate approach to control for inflated Type-1 errors of significance testing. There were two approaches to the quantitative analysis. First, because we had many variables that may differ between the two groups of participants, we used a multivariate analysis of variance (MANOVA). In a MANOVA one can compare whether groups of participants significantly differ in two or more quantitative variables of interest. Normal distribution of variables could not be upheld for many of the attitudinal items, however, given that there were in general no issues with outliers and homogeneity of variances, we employed parametric testing, as the general linear model approach (ANOVA/MANOVA) is usually robust to non-normality as long as the n is not small and relatively equal across groups, as in our case (Howell, 2009).

The second approach was a multiple regression approach to test which of a number of quantitative variables correlates significantly with an outcome variable. Here, we were interested to which degree variables such as length of employment, agile knowledge, team work satisfaction, quality of work environment, and quality of teamwork would correlate with (and therefore predict) overall satisfaction with the agile development process.

Analysis ASD satisfaction

To test whether the participant group (designer vs developer) affected the five dependent variables in the category "ASD-satisfaction" (ASD in team, ASD satisfaction overall in the organization, ASD satisfaction within respondent's product area, ASD fit with design goals, and to what degree the process could be improved) a 1-factor MANOVA was performed. Dependent variables were the five scores on satisfaction ratings as dependent variable (ranging from 1 to 7) with group (of participants) as fixed factor (Box's assumption of equality of covariance matrices was supported, p = .17). There was no main effect for the factor participant group, Wilks' λ = .984, F(4, 102) < 1 (see Table 4). Including Sex as an additional fixed factor (2-way MANOVA) did not change this result, Wilks' λ = .972, F(4, 100) < 1.

TABLE 4
Significant Univariate Effects for Group for the Questions on ASD Satisfaction

Source	Dependent Variable	Sum of Squares	df	Mean Square	F	Sig.
Group	ASD Team	2.894	1	2.894	1.853	.176
	ASD Overall	.904	1	.904	.543	.463
	ASD Product	1.406	1	1.406	.736	.393
	ASD Techniques	5.228E-5	1	5.228E-5	.000	.996
	ASD Improvement	1.153	1	1.153	.790	.376

Thus, 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 organization as a whole, and with what success (see Figure 1). Both roles thought that there was significant scope for improvements of the ASD process. Inspection of the means and post-hoc tests showed that both roles agreed that the ASD process could be improved, and overall satisfaction scores trailed scores for ASD satisfaction within people's own teams.

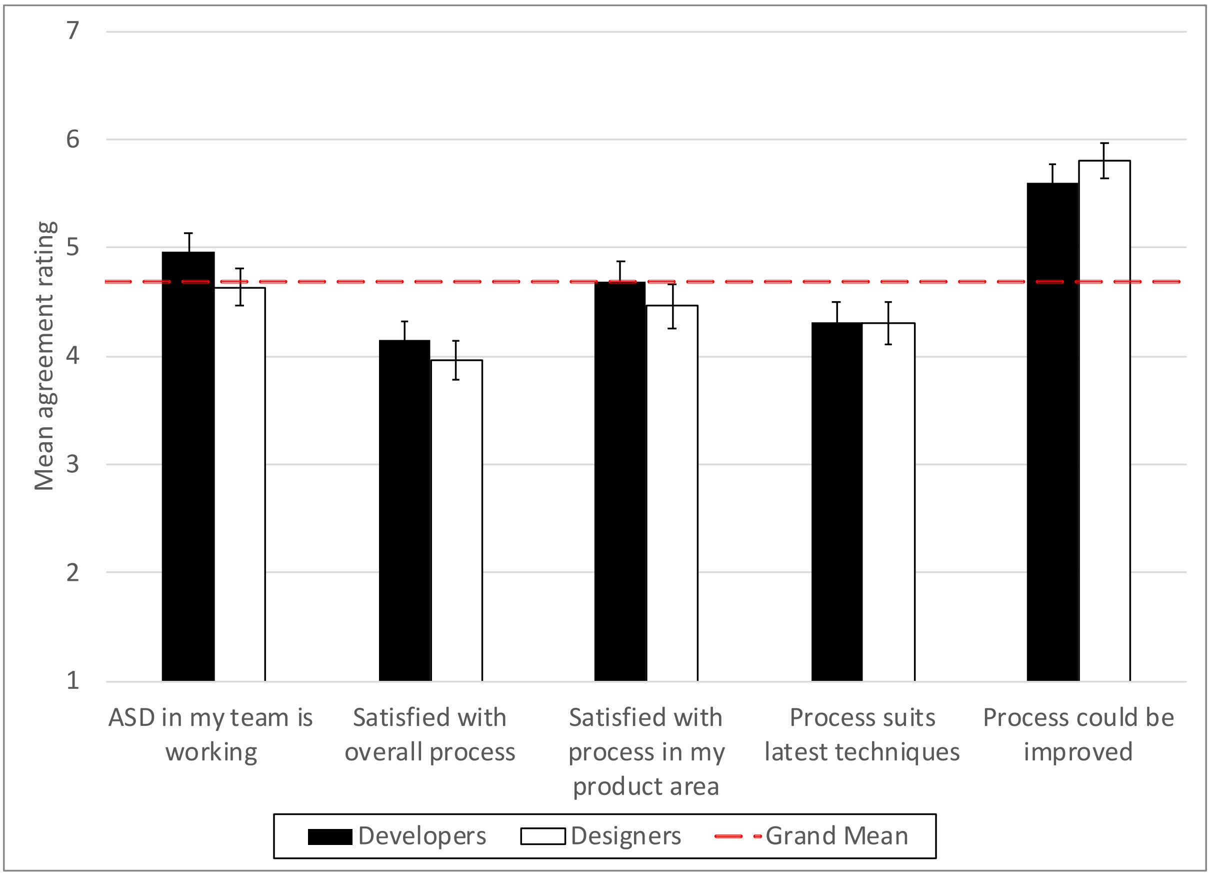

Figure 1: Participants in both groups responding "agree" or "strongly agree" to questions of satisfaction with ASD (see text for details).

Analysis of ASD teamwork

A further MANOVA was run with the items measuring teamwork (sharing ideas, having enough information, effort in problem-solving, team vision, Lurie et al., 2011) as independent variables. There was a main effect for the factor participant group, Wilks' λ = .902, F(4, 102) = 2.77, p = .031, partial eta2 = .098 (Box's assumption of equality of covariance matrices was supported, p = .12).

Follow-up ANOVAs (analysis of variance) for the factor participant group revealed a trend for an effect of the dependent variable shared team- information, F (1,106) = 3.91, p = .064, with developers perceiving to have more relevant information (M = 5.36, SD = .98) than designers (M = 4.98, SD= 1.12; see Table 5). Overall then, there was a significant difference on perceived quality of teamwork between designers and developers, with designers rating teamwork satisfaction significantly lower (and this difference was driven in particular by the variable 'perceived information sharing', and to a lesser degree by 'shared effort').

TABLE 5

Significant Univariate Effects for Group for the Questions on ASD Team Work

Tests of Between-Subjects Effects

Source	**Dependent Variable**	**Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
Group	Team Ideas	.023	1	.023	.023	.880
	Team Info	3.918	1	3.918	3.495	.064
	Team Effort	2.531	1	2.531	2.436	.122
	Team Vision	1.657	1	1.657	1.086	.300

In a separate analysis of one item - that did not fit directly into the teamwork scale, namely 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, t (105) = 1.95, p = .019; developers rate this higher (M = 4.27, SD = 1.56) than designers (M= 3.6, SD = 1.27). Figure 2 shows the pattern of results. Inspection of the rating means shows two major patterns: First, team ideas and team effort were rated rather highly, whereas satisfaction with team information and vision was significantly lower. Second, the quality of the ASD environment was the lowest rated item.

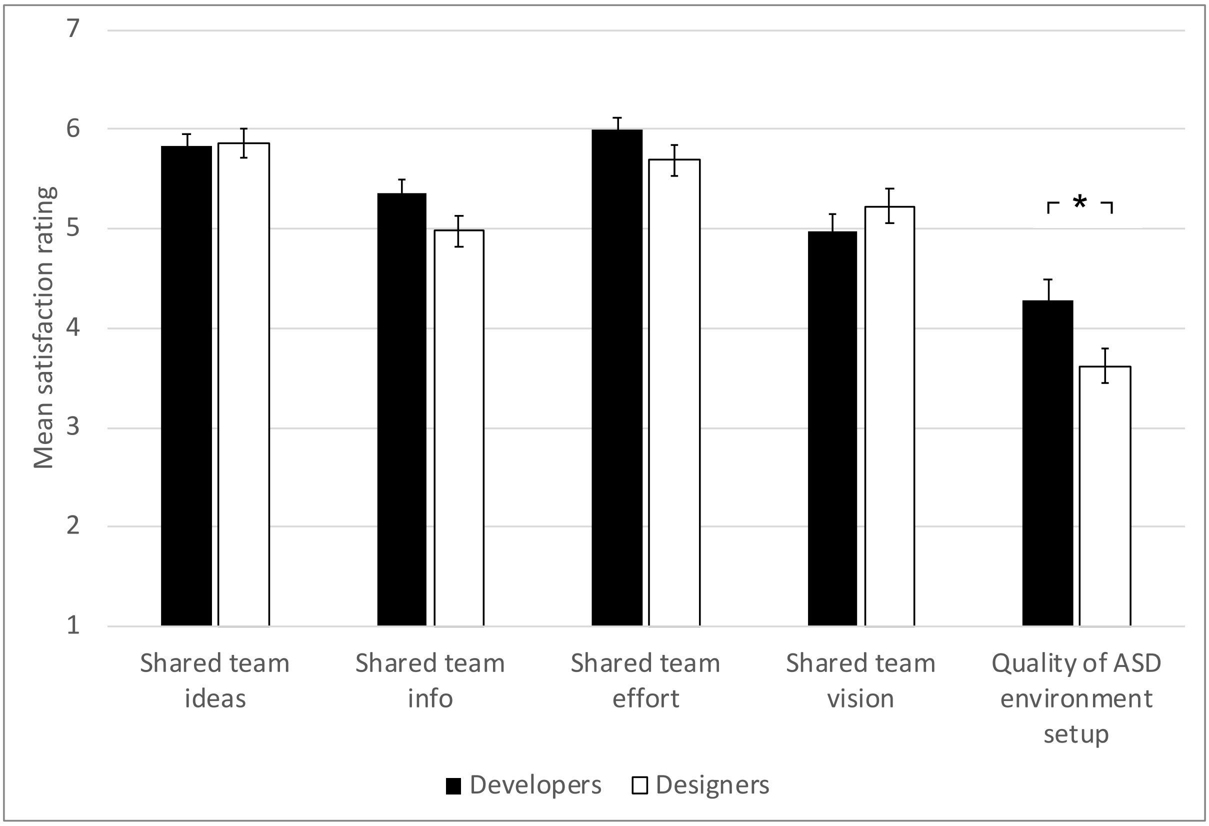

*Figure 2: Participants in both groups responding "agree" or "strongly agree" to questions of familiarity and level of perceived implementation of ASD. (Note: * = p <.05)*

Analysis cooperation and collaboration

Finally, a one-way MANOVA was planned with the variable group (of participants) on the set of questions asking about the perceived quality of cooperation and collaboration between designers and developers. However, Box's assumption of equality of covariance matrices was not supported, p = .005), Pillai's trace =.090, F(5, 101) = 2.77, p = .084, partial eta2 = .090. Therefore, Bonferroni-corrected multiple t-tests are reported instead of the follow-up ANOVAs (see Table 6).

TABLE 6
Inferential Statistics for the pairwise Group Comparisons (designers vs. developers) for the Questions on Collaboration.

Variable	**t**	**df**	**p**
Roles working together	-0.680	105.0	0.498
Roles relationship	-0.986	105.0	0.326
Roles Contributions	-1.945	105.0	0.054
Designer Skills in Develop.	-2.709	105.0	0.008 **
Developer Skills in Design	-0.649	105.0	0.518

Note. Student's t-test. * = p<.05; ** = p<.01; *** = p<.001

Multiple t-tests with the between-factor group revealed a marginally significant difference between ratings for the items perceived equal contributions, t (105) = 1.95, p = .05; developers rate this lower (M= 4.32, SD = 1.70) than designers (M= 4.90, SD = 1.33). Also significant was the difference in perception that designers use developer's skills, t (105) = 2.79, p = .008; developers rate this lower (M= 2.96, SD = 1.36) compared to designers (M= 3.69, SD = 1.42). However, these results are based on multiple comparisons, and only the difference in sharing skills would survive a Bonferroni-correction (with p = .01 as threshold for significance) and should be considered reliable. Figure 3 shows the pattern of results. Overall developers rated the statements significantly less positive than designers, F (4, 105) = 4.103, p = .045.

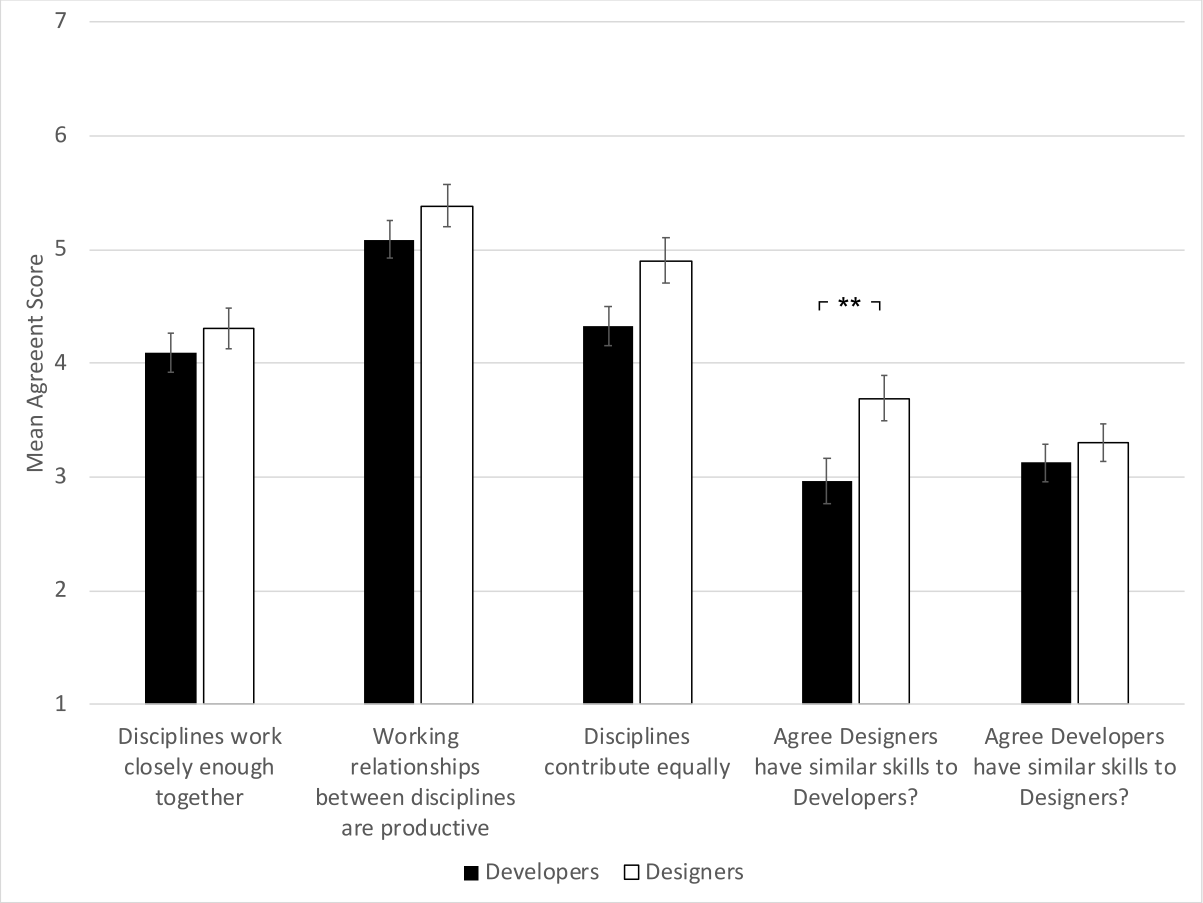

*Figure 3: Participants in both groups responding "agree" or "strongly agree" to questions of collaboration between Designers and Developers in ASD. (Note: ** = p <.01)*

Modelling the ASD satisfaction scores

In a final set of analyses, we employed a multiple linear regression analysis to investigate possible relationships between the tested variables (questionnaire items) in the two groups. This correlational approach was used to uncover possible associations between our variables that were not evident in the comparisons of means between groups.

First, we summarised the item groups into sub-scales by averaging scores (ratings). The average score on the questions on ASD (satisfaction and perceived level of implementation, Figure 1) was the dependent variable. The average score from the 4 questions on 'teamwork' (Lurie, 2011; Figure 2) was the first independent (predictor) variable, and the second variable was 'collaboration', which was the average score derived from the 5 questions on the specific working relationship between designers and developers (Figure 3). These clusters of questions were submitted to a reliability analysis and revealed Cronbach's alphas of > .70, which is sufficient to be used as a scale. Finally, we had scores from single items as further independent (predictor) variables: length of experience in organisation, knowledge about ASD, and perceived quality of the environmental setup at the work place (in regard to facilitating ASD, e.g. scrum boards).

These five variables plus a dummy-coded variable for role (designer vs developer) were entered into a multiple regression analysis. The resulting model significantly predicted a substantial amount of the variance, R square = .299; F(6, 104) = 8.380, p < .001). The significant predictors were environment, (designer and developer) collaboration, and quality of teamwork (see Table 7).

TABLE 7
Parameters and results for the multiple regression analysis.

Coefficients

Model		Unstand.	Error	Stand.	t	p
1	(Intercept)	0.923	0.808		1.143	0.256
	D&D Collaboration	0.338	0.085	0.349	3.973	< .001
	Teamwork	0.396	0.114	0.317	3.468	< .001
	Environment	0.149	0.059	0.216	2.499	0.014
	Familiarity w. Agile	-0.136	0.095	-0.137	-1.424	0.158
	Length at Org.	3.091e -6	9.943e -5	0.003	0.031	0.975
	Group	-0.287	0.188	-0.143	-1.527	0.130

Despite the lack of a group effect in the regression analysis, reflecting the lack of difference between designers and developers found in the first MANOVA on satisfaction scores, we ran additional separate regression analyses for designers and developers. This was based on the scale-level differences found for teamwork (MANOVA 2) and to other items as described above (environment, skills). In both groups the regression model predicted a significant amount of variance for the agile-process score: For developers, the adjusted R square = .295; F(5, 53) = 5.43, p < 0.001; for designers, the adjusted R square = .369; F(5, 50) = 6.84, p < 0.001.

Inspection of the multi-collinearity indicator (VIF) revealed that all values were acceptable (between 1.07 and 1.4) and that residuals were normally distributed. Significant predictor variables for the satisfaction with ASD in developers were the variables environment, beta = .266, p = .030, and (designer and developer) collaboration (combined questions), beta = .492, p < .001. For designers, the only significant predictor variable for ASD satisfaction was quality of teamwork, beta = .551, p < .001. Thus, whereas 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 striking because of the lack of overall statistically relevant group differences between designers and developers in their assessment and perception of ASD processes in the organisation (see 3.1.1 and 3.1.2). These points will be discussed in context of elicited comments from both respondent groups.

4.2.2 Qualitative analysis

The free-form comments participants gave in the survey (Typeform) were analysed separately using a qualitative approach. The comments were imported into a software tool for analysing qualitative data called NVivo (www.qsrinternational.com). We coded expressions of opinions, problems, events, reactions and interactions in the text by assigning the piece of text to a category (‘‘node’’ in NVivo). A category/node 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, we then used NVivo to create connections between categories and their subcategories. Following the analysis of the Qualitative data from the questionnaire, we grouped the 'phenomena’ into the following categories. These categories are described below with a selection of comments that supports each:

Theme category 1: Collaboration issues that impede ASD

Design work was perceived by both roles (designers and developers) 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.

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

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. The workflow in the current ASD was perceived as not iterative enough. In particular, it was felt (by developers) that design input was not provided early enough, which lead to inefficiencies. Despite this, developers reported that they would like to be more involved in the ideation phase and involved earlier in the process.

Theme category 2: Perceptions of other role’s working style

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.

Designers were perceived by developers to not know enough about the technical limitations when designing solutions. What this meant was that developers perceived inefficient and redundant work efforts.

Theme category 3: Perceived need for localised decision-making to enhance ASD process.

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 improve decision-making. Finally, there was frustration about the “sign-off” culture and decision-making in the organisation.

> There is also a major delay in decision-making - as in, this holds up projects because people seem reluctant to make decisions on things.

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.

4.3 Discussion of Study 1

The results of the survey and the qualitative analysis of the comments confirm some of the previously reported factors of successful ASD implementation, but additionally show which of these moderating variables vary across the two roles. There were significant differences in how designers and developers perceived the factors for 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.

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. Qualitative analysis suggests that these factors for agile satisfaction were for both groups associated with the frequency 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. These requirements are in contrast to the delays and alienation 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. 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 had no significant effect on moderating satisfaction with the development process.

The observation that we 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 our sample 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, our 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.

5 STUDY 2: PAIRING STUDY

Study 1 found both roles perceived collaboration in ASD contexts could be improved by specifically pairing the two roles, increasing co-location and thereby enhancing communication and co-operation early on in projects. Motivated by this finding, Study 2 further explored advantages and challenges of collaborative pairing between the two roles in UCD and ASD processes. To do this six pairs of UX designers and software developers across different teams in the organisation were approached for a Contextual Inquiry study. This was followed by a diary study over the course of six weeks.

5.1 Methods

The study took place within the same large organisation as described in section 3. Each of the pairs worked in different online product teams. The surrounding teams varied across the participant pairs, but all pairs were part of a multi-disciplinary product structure that included UX Designers, Software Developers, Business Analysts, Testers, Technical Architects, Project Managers and Product Managers. The data collection was split into two parts: A Contextual Inquiry (CI) in the form of semi-structured interviews at the beginning of the study, and a longitudinal study in the form of a participant diary study over the course of six weeks.

5.1.1 Participants

Designers were categorised (from their job titles) as "User-Experience Designers" and the developers were mainly "Software Engineers" or "Web Developers". To select the participant pairs, stakeholders (line-managers) from around the organisation were contacted and asked to suggest a range of individuals. From this list of eligible participants purposive sampling (Palys, 2008) was applied and the participants were selected. Out of the six pairs of participants, pairs A (A:Des and A:Dev) and B (B:Des and B:Dev) knew each other already, they had worked on projects together and were part of the same agile team. Pairs C (C:Des and C:Dev) and D (D:Des and D:Dev) knew each other, had worked together before on a project but were not in the same agile team. Finally, Pairs E (E:Des and E:Dev) and F (F:Des and F:Dev) did not know each other particularly well and they were organisationally placed into separate teams (but remained in the same broader product team). Their projects varied in terms of their aims, size and scope but these were comparably concerned with designing and developing web-based content. Whereas the developers were all attached to a single main project within each product the designers (except A:Des and D:Des) were spread across various projects.

5.1.2 Contextual Inquiry

To obtain rich information about work practices, the social, technical and physical environments a Contextual Inquiry (CI) was used with designers and developers (Wixon, Holtzblatt, & Knox, 1990). Using this method, the participants were first asked a set of pre-defined questions and then observed and questioned while they worked in their normal working environment. The focus of the interviews with the pairs was to openly discuss their previous experiences and their typical processes, tasks, opinions and their expectations for future. As part of the inquiry, interview sessions with the pairs took place as well as separate individual interview sessions. The aim was that participants could speak freely and talk about issues and barriers to success. It was stressed that the interview results would be kept as anonymous as possible. Table 8 below shows an example of the discussion guide that was used for the contextual interviews with the design and development pairs.

TABLE 8

Contextual Interview Sessions –Discussion Guide Questions

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?

5.1.3 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 Contextual Inquiry. 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. Participants were briefed during the Contextual Inquiry and were sent an email with essential information to take part in the study. A briefing document with the details for the diary entry template and examples was provided. The entries were collated on a Tumblr blog (which allowed entries via e-mail, smart phone or the blog website) which was password protected for each of them. 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.

5.2 Analysis

As described in section 5.1.1, 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. Although the researcher observed these factors, the main focus of the study was on the interactions between the designers and developers.

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 to create themes in the findings.

Following the analysis of the Contextual Inquiry and Diary Study, the themes from both data sets were analysed together in a triangulation approach; by using the 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).

5.3 Findings

The findings in this section are presented as follows: Firstly, the thematic analyses of the Contextual Inquiry and Diary Study are presented in tables 9 and 10 respectively. Each theme was categorised by the authors as falling in either the operational level of working in pairs, in teams, or regarding the whole organisation. The resulting themes from the combination of the two data sets are then described with a selection of supporting evidence.

TABLE 9
Contextual Inquiry – Table of thematic analysis findings

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 to follow a more data-driven method and audience research to inform joint decision making
	Separation of location caused frustration and led to less efficient ways of working practices	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 clarify concepts and define the increments of work	
Team	A lack of shared understanding of each other’s discipline	Breaking down tasks to enable making small decisions and iterations together	Desire to work together as a pair within a multi-disciplinary team - roles can cross over
	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 closely following Agile methodology

TABLE 10
Diary Study – Table of thematic analysis findings

Theme Categories	**Team collaboration & integration challenges**	**Pairing experiences and closer collaboration**	**Further opportunities for better integration**
Pairing	Ongoing separation between UX designers and Agile Developers	Pairing enables efficient and iterative collaboration to find solutions to work Location and close proximity is an enabling factor for close collaboration	Want more opportunities to sit in close proximity with each other 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 Paired work is seen to benefit whole process	Challenges of extending pairing method to more teams
Organisation	On going separation between UX designers and Agile Developers	Wider decision making is improved and more informed	

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 and integration challenges, Experiences of pairing and closer collaboration and Further opportunities for better integration. The themes from the study are then summarised in table 6.

5.3.1 Theme Category 1: Team collaboration and integration challenges

This section describes the collaboration and integration challenges that were experienced by the pairs of designers and developers.

Theme Category 1.1: Perception of operational separation between the two roles

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 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.

> 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.

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.

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, causing blame and tension between the two roles.

Theme Category 1.2: Lack of direct communication between the roles

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. Overall, communication between the other designers and developers was felt to be "lacking" and "could be improved".

> But of course, one problem is separating from each other […]. They cannot discuss their issues.

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. 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.

Theme Category 1.3: A lack of shared understanding of each other's discipline

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'). 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.

> 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.

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.

Theme Category 1.4: Separation of location and a lack of proximity

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. All pairs reported that being in separate locations was unsuitable to close collaboration (despite Agile routines and remote communication) and it resulted in little interaction. This was often out of their hands due to lack of space or the structure of their surrounding teams.

> 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.

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 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.

Theme Category 1.5: Frustrations of a lack of shared 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 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. 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.

> 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.

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. Despite their frustrations, 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.

5.3.2 Theme Category 2: Experiences of pairing and 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

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.

By sitting in close proximity to one another, either alongside each other or on nearby desks, it enabled more frequent communication and closer collaboration. Also, 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.

Theme Category 2.2: Early and frequent communication

Pairs found that direct communication early on in projects helped to reduce problems later on because of relationship-building and early sharing of ideas. Direct communication in person was preferred amongst the pairs because it would enable frequent discussions and ultimately better solutions to the particular work in question. Face-to-face communication was perceived by all pairs 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.

Regular communication helped to develop the pairs relationships and a mutual understanding of their roles - “it feels like a family […] with lots of banter”. It was acknowledged that if they communicated more, and thus developed a stronger relationship between each other, the team would be also be stronger. 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 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.

Theme Category 2.3: Co-creation of prototypes and in-browser style guides

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.

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. 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.

Theme Category 2.4: Shared ideation, experimentation and iteration

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.

> 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.

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.

Theme Category 2.5: Paired work was seen to benefit whole process

The designers and developers in the study perceived that pairing together on projects was beneficial to their design and development processes. 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.

> 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.

It was remarked that pairing helped to overcome "invisible" barriers between the two roles, allowing 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. 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.

Theme Category 2.6: Joint decision making

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 more informed decisions together. 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.

> 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.

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.

5.3.3 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

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.

> 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.

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. Despite this, 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.

Theme Category 3.2: Desire for more opportunities to sit in close proximity to enable collaboration

The pairs that were not permanently located together had a desire to more frequently sit in close proximity with one another, aiding deeper collaboration and "learning" between the roles. In particukar, 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.

> 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.

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.

Theme Category 3.3: More pairing and multidisciplinary team work where roles can crossover

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. Pairs talked positively about sharing the responsibility for the UX and how they could 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.

Pairs 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. Working with everyone together was felt to help in "producing the product" in a more collaborative way with a "better working flow"

Theme Category 3.4: Using more data and insights to inform decisions

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.

> 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.

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.

5.3.4 Summary of Pairing Study findings

The themes described above from the two data sets are summarised below in table 11.

TABLE 11
A summary of subsumed themes from Contextual Inquiry and Diary Study

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 multidisciplinary 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

6 DISCUSSION

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 Bhrel et al., 2015; Law & Larusdottir, 2015; Silva da Silva et al., 2011). Our current results extend previous findings by pinpointing the co-location and close collaboration between two crucial roles: Designers and developers.

This section summarises the findings from both studies and reflect them on the background of previous research literature. We then turn to how these inform future collaboration practices between UX designers and software developers working in an organisational setting, as well as directions for academic research in this area.

6.1 UCD and ASD Integration Challenges

In complex organisational settings, there are a number of on going challenges that have an impact upon the integration of UCD and ASD. Across both studies, 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 this, the views that emerge from the studies 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. However, there is also 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.

6.1.1 Organisational structure and team culture

Both studies demonstrate findings that the pre-defined organisational structure creates a separation and an "invisible" barrier between the UCD and ASD teams. In Study 1, the quantitative analysis of the survey responses show 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 the Pairing Study, 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: bringing about three main effects:

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). Secondly, the findings show that design work is perceived by both roles to be happening in too much in 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. Another outcome of this 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.

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 and provide a comprehensive view of the system or service being created, helping to reduce problems later on (Ferreira, Noble, & Biddle, 2007b). 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, the current evidence suggests that a large amount, or prolonged period of upfront design presents a significant challenge to collaboration. In this case, 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 subotpimal outcomes as reported previously by Ferreira et al. (2011) who suggest methods that foster closer collaboration and a more iterative approach to the design and development process.

6.1.2 Location and environmental setup

In addition to the organisational separation of the UCD and ASD team(s), physical separation was also 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 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 analyses of the responses suggest that these factors for Agile satisfaction were associated with the level of pairing of roles and increased physical co-location. Similarly, in the Pairing Study, 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 often to be out of their control 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 moving to just another floor or another area of the same floor in the building significantly disrupts frequency and the quality of teamwork.

The current findings therefore support the idea 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). Similarly, 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. Alongside our findings, 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, our 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 these 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, this contrasts with the desires of the designers and developers in the Pairing Study 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.

6.1.3 Decision-making

The designers and developers confirmed that decision-making processes are a challenge in both the Online Survey and the Pairing Study, reflecting reports elsewhere (e.g., Drury-Grogan & O'Dwyer, 2013). The online survey respondents perceived senior stakeholders' hierarchical decision-making process as a barrier to successful ASD. In the Pairing Study the nature of a large organisation was thought to hamper decision-making and autonomy of the roles on localised level due to the requirements of working with multiple teams and dealing with dependencies. Designers also showed frustration about the lack of involvement from their senior stakeholders partly because of their back seat approach and their reliance on agile work practices, while at the same time traditions and organisational requirements asked them to sign-off and make key decisions about the work. However, because of senior stakeholders' lack of involvement and lack of knowledge of agile work that had gone into design and development work already, this slowed-down development progress and caused frustration among designers with senior stakeholders' decisions. The developers 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, yet cross-disciplinary and autonomous decision-making.

Overall, the findings support the view that agile decision-making at a localised level between designers and developers is a challenge 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 in ASD remains a challenge for many organisations either because they are ignoring or lacking adequate decision-making processes (Highsmith, 2009).

6.2 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 Study 2 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.

Previous work has identified and discussed the challenges and solutions of Agile and UCD integration (see Introduction, Table 1). A recent extensive review by Brhel and colleagues (Brhel et al., 2015) has summarised over 83 relevant publications in this area and derived five principles of successful user-centered agile software development (UCASD). These principles are: 1. Separate product discovery and product creation; 2. Iterative and incremental design and development; 3. Parallel interwoven creation tracks; 4.Continuous stakeholder involvement; 5. Artifact-mediated communication (see Brhel et al., 2015, for details). In Table 12 we map these principles - together with previously suggested solutions to Agile/UCD challenges – onto our findings from Study 1 and 2. However, the authors point out that they could not derive a principle for the people/social dimension, as contradictory evidence for the effectiveness of cross-functional teams versus separate design and development teams was found. Our suggested mapping of paired work adds therefore the social (teamwork) dimension to their five principles. In addition, recall that the survey found 3 main predictors for ASD satisfaction that likely have different impact on these principles: Quality of teamwork (principles 2, 3, and 4), quality of designer and developer collaboration (all principles), and quality of (ASD) environment (principles 1, 2, and 5). Experience with ASD methods was not a factor, unlike found in previous work, which may reflect our findings that it is the quality of experience that counts, rather than mere exposure.

TABLE 12

Principles & previous solutions for ASD and UCD integration as applied to the current observation of paired practice

Principle Description (UCASD, Brhel et al., 2015)	Previous Solutions (see Table 1)	Factors for success between designers and developers
1. Separate product discovery and product creation	Separate pre-development phase (sprint 0) / "upfront design"	Early and frequent communication; Shared ideation and problem solving;
2. Iterative and incremental design and development	Flexible chunking (or time-boxing) of design activities; well-defined design goals	Crossover of knowledge & skills; Making joint decisions;
3. Parallel interwoven creation tracks	Communicating design vision as early and frequently; ad hoc meetings	Early and frequent communication; Co-creation and prototyping; Close proximity;
4. Continuous stakeholder involvement	Discount usability evaluation; UX specialists as surrogate users or user interface	Making joint decisions; Close proximity; Co-creation and prototyping;

	inspectors	
5. Artifact-mediated communication	Artifact-based/ web/wiki-based documentation; frequent interaction	Co-creation and prototyping; Early and frequent communication

6.2.1 Close proximity is key

The pairing study between designers and developers found location and close proximity to be an crucial factor for close collaboration and is a significant facilitator for effective communication, regular ad-hoc discussions and reducing handovers between roles. Equally, 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 shaped cooperation views upon the roles.

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 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).

6.2.2 Early and frequent communication

Through the pairing study, 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. Study 2 shows that this can be mitigated somewhat by involving the developers early in the design process, as advocated by Salah et al. (2014). 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, Noble, & Biddle, 2007a). 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 (see also Kollman, 2008).

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 that an understanding of the overall vision and direction quickly breaks down – and is even reported as being useless by Kollman (2008). 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.

6.2.3 Shared ideation and problem solving

Study 2 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 particular, it reduces the 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 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 user 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 finding shows 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.

6.2.4 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 they 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 high level of more frequent knowledge sharing that is found to occur through pairing helps to breaking down 'invisible' barriers between roles and bridge separations, allowing the disciplines to rightfully crossover and overlap 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.

6.2.5 Co-creation and prototyping

A key and unexpected finding from the pairing study was the shared creation of prototypes between some 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. In particular, Ungar (2008) and Sy (2007) suggest that designers work one iteration ahead of developers regarding prototyping and according to Chamberlain and Sharp (2006), they should be willing to "feed the developers" with prototypes. Despite this, 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 collaborating on the prototypes, they also had produced something that was closer to production than usual 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, our analysis showed that 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'. This high degree of collaboration between the pairs in producing shared artefacts is a further advantage because it can serve the wider team or other projects 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 our results, 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 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..

6.2.6 Making joint decisions

The pairing study found that success between designers and developers comes by 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. The importance of the decision-making processes reflects reports elsewhere (e.g., Drury-Grogan & O'Dwyer, 2013) and show that decision-making is an on-going challenge that requires support, both at a managerial and a localised level in an organisational setting. There appears to be a strong desire from the designers and developers to improve this, providing more ownership to teams and reducing the risk-aversive culture found in the organisation. In the pairing study in particular, a successful localised method was found through making iterations with quick collaborative decision-making in pairs as the component (e.g. a responsive web component) was developed in the browser. These findings highlight that collaboration is core to an Agile team (Beck et al., 2001). Agile teams self-organise and are meant to contribute collaboratively to make decisions (Nerur et al., 2005; Schwaber & Beedle, 2001).

7 STRENGTHS AND LIMITATIONS

This investigation employed a mixed methods approach to study the success and barriers of Agile processes from the point of view of designers and developers, in particular in a large organisation. It meant that objective survey results pointed to significant yet nuanced differences between designers and developers in their perception and satisfaction of agile methodology, which were confirmed by qualitative methods. In addition, contextual inquiry and diary study results identified clear reasons and context for the success and barriers regarding agile implementation in a large organisation. Thus, the current studies provided a large number of in-depth insights usually not easily generalisable to a larger population. However, there are of course limitations. The findings appear at first not to be readily applicable to other environments (or organisations), which is often the nature of applied research and the type of sampling one is restricted to. 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, our 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 comparable situation as the observed environment in at least one 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 aspect not addressed here is a psychological type of barrier in the working relationship between designers and developers: 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.

There are further limitations, for example, we did not differentiate between different types of ASD practices (e.g., see Law et al., 2015) or management styles (Dyba, 2008). Future research may therefore aim to address how senior stakeholder's adoption of agile philosophy and the associated need to relinquish decision-making powers influences collaboration and cooperation in ASD teams. In particular, it would be interesting to track where, how and when decisions are taken in the development process and measure their quality and outcome.

8 CONCLUSION

The findings in this report support the idea 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 two empirical studies conducted in a large UK based organisation. The research literature often argued that attempts to integrate UCD and ASD are heavily influenced by practice but few recent studies have investigated the settings and contributing factors of practitioners' work. In addition, a recent review by Caballero et al. (2016) showed that the majority of UCD roles work independently from developers, at least in early stages of projects.

Study 1 found that both roles perceived that collaboration can be significantly improved. For satisfaction with ASD processes were environmental setup and collaboration with designers, while for designers the main predictor was the perceived quality of the (wider) teamwork. A novel finding was that these factors were not moderated by ASD knowledge or experience. Follow-up qualitative analyses confirmed these results. The traditionally suggested interventions of training and mentoring were not a priority for our sample, but rather identified the importance of close collaboration between the two roles of developer and designers. In addition, ASD processes were perceived to be hampered by top-down decision-making and a persisting "sign off" tradition in the organisation.

In conclusion, and in answer to our research questions, this study finds 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 findings from previous work (e.g., see reviews by Bhrel et al., 2015; Law & Larusdottir, 2015; Silva da Silva et al., 2011) and have important implications for practitioners and researchers.

9 REFERENCES

Albisetti, M. (2010). Launchpad's quest for a better and agile user interface. In *Lecture Notes in Business Information Processing* (Vol. 48 LNBIP, pp. 244–250). https://doi.org/10.1007/978-3-642-13054-0_26

Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. *XP Series*. https://doi.org/10.1136/adc.2005.076794

Beck, K., Beedle, M., Bennekum, A. Van, Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved from http://agilemanifesto.org/

Beyer, H. (2010). User-Centered Agile Methods. *Synthesis Lectures on Human-Centered Informatics*, *3*(1), 1–71. https://doi.org/10.2200/S00286ED1V01Y201002HCI010

Bias, R. G., & Mayhew, D. J. (2005). *Cost-justifying usability: an update for an Internet age*. *Alaska Fisheries Data Series* (Vol. Second). https://doi.org/ISBN-10: 0120958112

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Brown, D. D. (2013). Five Agile UX Myths. *Journal of Usability Studies*, *8*(3), 55–60.

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

Budwig, M. (2009). When User Experience Met Agile : A Case Study, 3075–3083.

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.

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

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. A. D. (2002). Usage-centered engineering for Web applications. *IEEE Distributed Systems Online*, *3*(3). https://doi.org/10.1109/52.991331

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*(6), 1097–1120. https://doi.org/10.1142/S0219622013400105

Dyba, T., & Dingsoyr, 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

Ferreira, J., Noble, J., & Biddle, R. (2007a). 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. (2007b). 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. (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

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

Gothelf, J., & Seiden, J. (2013). *Lean UX. LEAN UX*. https://doi.org/10.1017/CBO9781107415324.004

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

Highsmith, J. (2009). *Agile Project Management: Creating Innovative Products*. *Management*.

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

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

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

Lee, J. C. (2011). Evaluating eXtreme Scenario-based Design in a Distributed Agile Team, 863–877.

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

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

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

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., & 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

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

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

Nielsen, J., & Norman, D. (2015). The Definition of User Experience. Retrieved from http://www.nngroup.com/about-user-experience-definition

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

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

Schwaber, K. (1997). SCRUM Development Process. In *Business Object Design and Implementation* (pp. 117–134). https://doi.org/10.1007/978-1-4471-0947-1_11

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., & 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*, *5*(10), 743–751. https://doi.org/10.4236/jsea.2012.510087

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

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

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

10 Appendix

10.1 Questions in survey (Study 1)

Appendix: Question in online survey (Study 1)

General:

- How long have you worked in your current role for?
- I am familiar with working in an iterative development process (e.g. Agile).

Section ASD satisfaction:

- Overall, the design and development process in my team (Workstream or Product area) is working in an agile way.
- Overall, I am satisfied with the current design & development process within the BBC in general.
- Overall, I am satisfied with the current design & development process within the BBC in my product area.
- Overall, I think the current design & development process suits the latest techniques such as Responsive Web Design.
- Overall, I think that the current design & development process within the BBC could be improved.

Section team work:

- My team encourages everyone to share ideas.
- People in my team have the information that they need to do their jobs well.
- When people in my team experience a problem, they make a serious effort to figure out what's really going on.
- Everyone in my team feels able to act on the team vision.
- Overall, I think the environment we work in (e.g. shared work spaces, Jira boards, etc.) is set up appropriately for our work.

Section designer-developer collaboration:

- Designers and Developers work closely enough together.
- Working relationships between Designers & Developers are productive (e.g. they are responsive to suggestions, communicate efficiently, etc.)
- Designers and Developers contribute equally to the design development process for BBC sites/products
- Designers employ a set of skills (e.g. writing code) that are similar to the ones used by Developers.
- Developers employ a set of skills (e.g. ideation, wireframing) that are similar to the ones used by Designers.

10.2 Supplementary tables

Supplementary TABLE 1
Descriptive statistics for the variables in Study 1.

	Group	N	Mean	SD
ASD Team process	SEngineer	55	4.964	1.217
	Designer	52	4.635	1.284
ASD Overall	SEngineer	55	4.145	1.224
	Designer	52	3.962	1.357
ASD Product	SEngineer	55	4.691	1.289
	Designer	52	4.462	1.475
ASD Techniques	SEngineer	55	4.309	1.451
	Designer	52	4.308	1.449
ASD Improvement	SEngineer	55	5.600	1.196
	Designer	52	5.808	1.221
Team Ideas	SEngineer	55	5.836	0.898

	Group	N	Mean	SD
	Designer	52	5.865	1.085
Team Info	SEngineer	55	5.364	0.988
	Designer	52	4.981	1.129
Team Effort	SEngineer	55	6.000	0.882
	Designer	52	5.692	1.147
Team Vision	SEngineer	55	4.982	1.254
	Designer	52	5.231	1.215
ASD Environment	SEngineer	55	4.273	1.569
	Designer	52	3.615	1.270
Roles working together	SEngineer	55	4.091	1.724
	Designer	52	4.308	1.566
Roles relationship	SEngineer	55	5.091	1.531
	Designer	52	5.385	1.549
Roles Contributions	SEngineer	55	4.327	1.700
	Designer	52	4.904	1.332
Designer skills in Developm.	SEngineer	55	2.964	1.360
	Designer	52	3.692	1.422
Developer skills in Design	SEngineer	55	3.127	1.415
	Designer	52	3.308	1.462

Authors

Alexander Jones –University of Chester & BBC – alexander.jones@chester.ac.uk

Alexander Jones is a designer at the BBC and a Computer Science PhD student at the University of Chester. His current interests are in the areas of multidisciplinary collaboration and ethical design and in particular how technology can be designed to help and support with children’s online wellbeing.

Dr Volker Thoma – University of East London - v.thoma@uel.ac.uk

Dr Volker Thoma is a Reader in Cognitive Psychology at the University of East London, after a postdoctoral position at UCL, and a PhD in Psychology (Goldsmiths London). His interests are mainly in the areas of attention, recognition of objects and consumer items, human factors, and judgment and decision making.