

UX Design in Agile: A DSDM Case Study

Laura Plonka¹, Helen Sharp¹, Peggy Gregory², and Katie Taylor²

¹ The Open University, Walton Hall, Milton Keynes MK7 6AA, UK

² University of Central Lancashire, Preston PR1 2HE, UK

{laura.plonka, helen.sharp}@open.ac.uk,

{ajgregory, kjtaylor}@uclan.ac.uk

Abstract. Integrating User Experience (UX) design with agile development continues to be the subject of academic studies and practitioner discussions. Most of the existing literature focuses on SCRUM and XP, but in this paper we investigate a technical company who use DSDM. Unlike other agile methods, DSDM provides a configurable framework and a set of roles that covers the whole software development process. While elements of the UX design integration experience were similar to those reported with other agile methods, working practices to mitigate the challenges were identified using DSDM's standard elements. Specifically, communication challenges were mitigated by extending two of DSDM's standard roles. In addition, a change of focus between a design-led phase and a development-led phase of the project changed the communication challenges. Agile teams need to be aware that this change of focus can happen and the implications that it has for their work.

Keywords: DSDM, UX, agile roles.

1 Introduction

Adequately addressing the user perspective is critical for software system success [1], and good user experience design is fundamental to achieving this. How best to integrate user experience (UX) design into an agile project has been a concern of practitioners and researchers for many years [2, 3, 4]. The main agile methods do not provide robust support for this integration, leading to several experience reports and much debate in the UX community. Several approaches to integration have been suggested including aligning processes, utilising UX techniques alongside agile sprints, and co-location of experts. However challenges remain.

UX design is about designing “how the product behaves and is used in the real world... how it works on the outside, where a person comes into contact with it and has to work with it.....every product that is used by someone has a user experience: newspapers, ketchup bottles, reclining armchairs, cardigan sweaters” [5]. This involves producing Wireframes, visual designs, interface widgets, user characterisations, and performing user research and usability testing.

In this paper we present a case study that explores the challenges faced by a company when integrating UX design into a Dynamic Systems Development Method (DSDM) project. We use an iterative research approach because it enables us to work

closely with an organisation on challenges that are relevant to everyday practice. Dingsoyr et al [6] have called for more research that has industrial impact to build a body of knowledge about agile methods that is relevant to practitioners. DSDM is of interest because it has been studied less intensively than other agile approaches [7], yet it provides a particular perspective on project phases and team roles that sets it apart from other agile methods. The case study focuses on a high-tech software development company that has a core expertise in software delivery and subscribes to the DSDM method. This case study presents mainly the technical team's perspective since it was they who perceived the difficulty.

The main research question of this study is "What challenges are faced by a company trying to integrate UX design and DSDM agile development?" Addressing this question also allows us to discuss two follow-up questions "How does the DSDM framework support this integration?" and "What implications do the answers to these questions raise for DSDM and other agile methods?"

The paper is structured as follows: section two introduces the DSDM framework; section three summarises previous work on integrating agile and UX design; section four introduces the study site, and describes the research approach; section five presents our findings; and section six revisits the research questions.

2 The DSDM Framework

The Dynamic Systems Development Method (DSDM) is an agile framework for both project management and product delivery that grew out of the Rapid Application Development (RAD) tradition (www.dsdm.org). It was the earliest published agile method, and one of the founders, Arie van Bennekum, was an original signatory of the Agile Manifesto [8]. In contrast to early versions of XP and Scrum, which focused on engineering practices, DSDM sought to wrap the best aspects of RAD in a lightweight framework to ensure the delivery of business value. The method has evolved into the DSDM Agile Project Framework [9]. More details about how the different agile methods compare can be found in [7].

The DSDM framework covers the full project lifecycle including guidance on philosophy, principles, project roles, processes, practices and products. It is typified by key practices such as iterative and incremental development, MoSCoW (Must, Could, Should, Won't have this time) prioritisation, Timeboxing, Modelling, Prototyping and Facilitated Workshops. It is configurable so it can accommodate a range of project types and sizes; and is compatible with a variety of governance and programme office structures [9].

In this paper we give examples from DSDM Atern [9] because this version was used within our case study organisation. Atern consists of seven phases: Pre-project, Feasibility, Foundations, Exploration, Engineering, Deployment, and Post-project. In Pre-Project a proposal is formalised and prioritised in line with strategic goals. During Feasibility the business and technical viability of the project are considered. A high-level investigation of potential solutions is produced, as well as estimates of costs and timeframes. During the Foundations phase business needs are ascertained and high level requirements are identified, prioritised and linked to those needs, and resources are secured. Planning allocates high level requirements to increments

(releases). Each increment consists of a number of smaller development timeboxes, the size of which is decided by the team. The first three phases are sequential and set the scene before the actual development begins. For each increment, Exploration and Engineering iteratively investigate solutions through the development of prototypes that build, test and document the solution. In Deployment the solution is made operational. The number of passes through this phase will depend on the number of increments scheduled and will be driven by business need. The Post-Project phase takes place after the last Deployment phase. It is used to assess project performance against business value and determine benefit realisation.

DSDM defines a full set of roles for project teams: Business Sponsor, Business Visionary, Technical Coordinator, Project Manager, Team Leader, Business Analyst, Solution Developer, Solution Tester, Business Ambassador, Business Advisor, Workshop Facilitator and DSDM Coach. The framework outlines the responsibilities required however these roles are filled in different ways depending on the nature of the project, but a key aspect is the importance placed on business involvement.

3 Integrating Agile Development and UX Design

Approaches to integration have been reported by both practitioners and academic researchers. They can be broadly divided into two categories: bringing people together, and aligning developer and UX designer work practices.

3.1 Bringing People Together

Cross-functional, co-located teams are regarded as imperative for agile to work. For example, a key practice in the XP agile approach is the ‘whole team’ practice. However this view is problematic. Firstly, it is often not feasible or desirable to co-locate UX designers and agile developers. For example because the organisation’s core business does not support the direct employment of UX designers, or where the organisational culture keeps the disciplines separate: “UX designers work best when they are separated from the issues of software construction because these issues hamper creativity” [10].

Secondly, relying on cross-functional teams assumes that bringing people together leads to the integration of concerns, but does it? Ferreira et al. [11] found that integrating UX design and developers is an ongoing achievement, requiring articulation work and conscious effort day-by day, so although co-location helps, it is not the whole story. They identified four aspects to this: integration as mutual awareness; integration as negotiating progress, integration as engaging with each other and integration as expectations about acceptable behaviour.

3.2 Aligning Developer and UX Designer Work Practices

Agile developer and UX designer work practices may be aligned in a range of ways such as: using techniques from one discipline in the other, combining agile and UX design processes, and recommendations derived from practice.

Techniques from UX design such as personas [12, 13], discount usability [14] and scenarios [15] have been reportedly used within agile projects. Personas can act as reminders to developers about who they are developing for, and hanging posters of personas in the development team area can make design work more visible to developers [13]. Looking at it the other way around, Kollmann et al. [16] describes the idea of a “Question Board”, which is similar to an agile progress board but focuses only on design issues. They explain that it facilitates and triggers discussion about open questions and issues related to design. It also helps to avoid recurring debates and captures different perspectives. Sy [17] also suggests capturing design issues as story cards on a UX board to increase their visibility.

UX design has traditionally followed a process that includes big design up front – something that agile tries to avoid. Aligning these processes can therefore be a challenge. In response to this, Sy and Miller [18] proposed that UX designers work one timebox ahead of developers (see Figure 1), which has become very popular. This enables the design work to be completed ahead of development work yet be tightly coupled to it, as the user stories evolve.

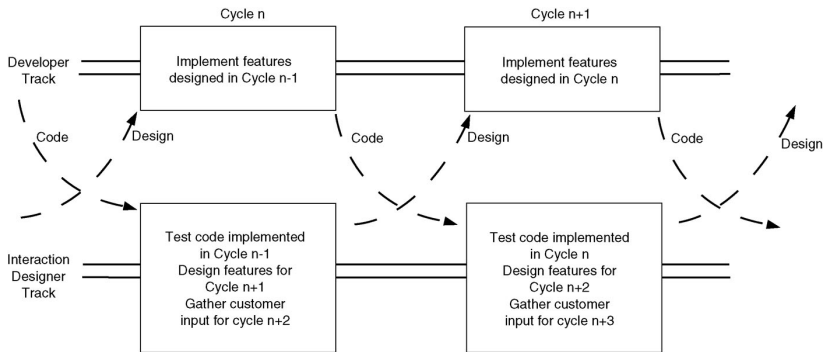


Fig. 1. ‘Train tracks’ development where UX designers work one timebox ahead of agile developers [15]

Several sets of recommendations have been developed by practitioners. For example, Jeff Patton has 12 recommendations for successful UX design integration [19] including “Research, model and design up front, but only just enough”, and “Buy design time with complex technical stories”. Nielsen and Norman [20] recommend development in train tracks, and emphasize the need to maintain a coherent vision.

4 Research Approach

This paper is based on a case study conducted by the Agile Research Network (ARN – see agileresearchnetwork.org), a network that conducts industry based research into agile methods. LShift, a hi-tech software development company, approached the ARN to investigate a challenge that they were facing: Integrating UX design into a DSDM project. The research was carried out between April 2013 and October 2013 using an

iterative research approach that incorporates regular feedback points in which observations and findings were presented back to the development company [21]. This research approach was chosen based on two main considerations:

1. The research was conducted on a project that was running at LShift at that time. Studying a live project means that work practices and challenges that the team members face are constantly evolving during the course of the project and hence requires an iterative data gathering process to keep up to date with the changes.
2. LShift approached the ARN with a real-world challenge and the aim of this research was to generate research insights to address the research questions at the same time as providing research that helps practitioners [22]. In the context of this study, this means that we shared observations, findings, and relevant literature during the course of the case study.

4.1 Research Site

LShift is a hi-tech software development company that works across a broad range of industries, languages and platforms. They have tried many flavours of agile and subscribe to the DSDM method. At LShift all employees are co-located in an open-plan office. However due to the wide variety of projects they produce, the expert LShift software engineers regularly work with external experts and additional teams such as partner agencies or client-owned teams who are often not co-located for an entire project. This was the case for UX experts. LShift did not employ UX designers themselves. Instead their UX design work was done by a separate UX design agency that had some experience of agile but did not specifically subscribe to the DSDM method. The agency is a separate commercial entity, located in a separate building.

4.2 Data Gathering and Analysis

Our research approach consisted of four data gathering phases, two main feedback points and a jointly written report by the researchers and the company. Figure 2 shows a simplified timeline of our research approach. In practice, phases were iterated.

1. **First data gathering phase: Initial interview** At the beginning of the case study, three researchers interviewed the managing director and the programme manager to build an overall understanding of the project and to develop an initial picture of the as-is situation in the project and the perceived challenges. This interview served as a starting for the following observations.
2. **Second data gathering phase: Observations** During the case study, two researchers spent time at the LShift office observing the daily work practices of the development team and attending meetings. Table 1 presents an overview of the meetings attended. In addition, the two researchers had informal conversations with developers and project managers during lunch and coffee breaks. During all observations, extensive field notes were taken.

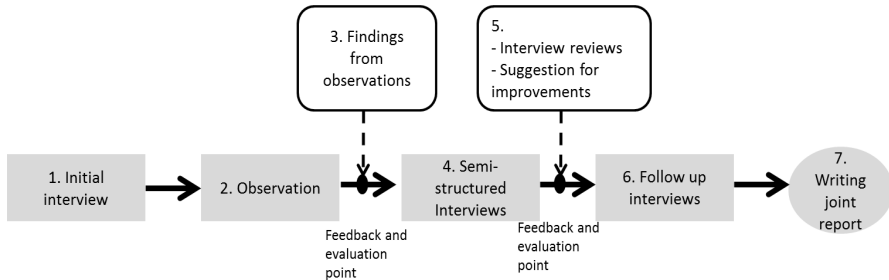


Fig. 2. Research Approach

- 3. First feedback point:** At the first feedback point, initial observations and identified challenges were shared with LShift’s management. The management supported us in identifying key stakeholders for interviews.

Table 1. Overview of observed meetings

# Observations	Type of meeting
5	Daily Stand-Up Meetings
1	Planning and estimation session
2	Retrospective (one with designers and one without)
1	Design Review Session

- 4. Third data gathering phase: Semi-structured interviews** Semi-structured interviews were conducted with the identified key technical figures (developers, project managers, and a business analyst) and UX designers but subsequently the designers withdrew. The interviews were used to delve deeper into the identified challenges, to uncover additional challenges and to understand the perspectives of different key stakeholders. All interviews were audio-recorded and conducted by at least two researchers; each one lasted between 30min and 1 hour.
- 5. Second feedback point: Presentation** Before the second feedback point, we (all four authors) analysed the interviews to understand the different perspectives on the challenges. We also reviewed existing literature to identify ways to mitigate these challenges. Subsequently, findings from the analysis and literature review were presented in a meeting that was attended by a large proportion of LShift’s project team. In the meeting, project members confirmed the challenges we identified and discussed how to mitigate them using the literature review.
- 6. Fourth data gathering phase: Follow up interviews** To keep track of the changes that took place after the second feedback point, follow-up interviews with the project manager and the managing director were conducted.
- 7. Joint written report** As a final step of our research approach, a joint report with the company was written [23]. This final step allowed us to validate the interpretation of our observations.

The analysis of the data was conducted iteratively with a focus on articulating integration challenges. After each data gathering step, the data was analysed through group discussions, key themes were extracted and findings confirmed with the company. The findings of the each step informed further data gathering.

5 Findings

LShift recognized early on that developers and UX designers have different perspectives and goals, different processes, different commercial pressures, and different skills and knowledge. These differences manifested themselves during the “Feasibility and Foundations” phase and the “Engineering” phase. The next two sections present how UX design was integrated into DSDM during that phase, what challenges the project members were facing and how they addressed them.

5.1 Integrating UX Design during Feasibility and Foundations

For the project team the journey began at a very early stage. While the concept and business case for the new product was being developed, a number of user interface concepts were prepared which became a key part of the sales presentation. While it was understood by all that these were for illustrative purposes to bring the concept to life, they did create a set of expectations about the scope of the project and these expectations survived through the planning stages.

5.1.1 Working Practices during Feasibility and Foundations

From the beginning of the Feasibility planning stage, UX designers and developers together ran workshops to explore user journeys, produce a high level picture of what the product had to do and estimate the size of the design and development effort.

At the beginning of the Foundations phase, developers and designers were working mainly independently. LShift, the development company, who provided all the technical expertise and is the company with delivery responsibility, focused on fleshing out the high-level user stories, the technical analysis, infrastructure and architecture, the security design, and technical de-risking. Meanwhile, the designers created UX concepts and personas and collaborated closely with the client suggesting and deciding on designs without receiving technical input from the developers. The suggested designs were accepted by the client and consequently set their expectations.

5.1.2 Challenges during Feasibility and Foundations

LShift faced two main challenges during Feasibility and Foundations:

1. Technical feasibility issues with design-led approach: To focus on the potential features of the product unencumbered by practicalities of having to deliver them, a design-led approach was chosen for the early stage of the project. Illustrating potential features using interface design mock-ups can be a very useful tool for providing a shared understanding of what's to be built. However, issues with this approach arose because UX designers and developers worked mainly independently, and designs were agreed with the client before developers confirmed their technical feasibility. This resulted in challenges because the client's expectations had been set, but some elements of the design had to be changed after developers discovered that they were not feasible.

2. Agile prioritization: The prioritisation of functionality in agile projects meant that the implementation of some features was delayed to a later increment than originally planned, and some features moved between timeboxes.

Whether the functionality changed because of technical feasibility issues or prioritisation activity, the result was the same: the client did not receive the functionality envisaged in the design illustrations, and this was a challenge.

5.1.3 Mitigations during Feasibility and Foundations

In order to mitigate this challenge, a developer was seconded to the UX team to work in a **Business Analyst role**, initially with a view to spend time with the design team and help assess the feasibility of design proposals earlier in the process. However, the initial planning of resources did not account for this extra, time-consuming task for the technical lead. This resulted in an extended Foundations phase because the technical tasks did not progress as quickly as expected.

5.2 Integrating UX Design during Engineering

During Engineering, the developers' work was broken down into increments comprising a number of three-week long time-boxes. The team held daily stand-up meetings and organised their user stories in a project management tool. The UX designers' work was also time boxed and they worked one sprint ahead of the development team (as shown in Fig 1). The designers organised their work during their timeboxes independently of the development team.

5.2.1 Working Practices during Engineering

During Engineering, developers and UX designers had to integrate their work with each other, organize the hand-over of designs and feedback on the designs.

The UX designers worked closely with the client through an iterative design process resulting in UX designs that were usually signed off by the client before being handed over to the development team. The developers received various documents. Usually designs were detailed, "pixel perfect"¹ and signed off by the client. However, occasionally developers received interactive wireframes. A design review to verify the implemented designs was conducted at the end of each increment.

Due to stakeholder constraints, the classic DSDM whole-team workshops were not run, but other communication-focused activities were in place.

Daily Communication: Designers attended the daily development stand-up meetings (either in person or on the phone). This provided a daily opportunity for communication and keeping up to date with each others' work.

The New Business Analyst (BA) Role: Identified during the Foundations phase this role continued to be a "communication bridge" between the developers and the client as well as between the developers and the design agency. Over time, this

¹ Pixel perfect design is the process of aligning and sizing all of the objects that make up a design to their exact pixel placements and sizes.

communication bridge worked more directly with the client to drive the business requirements and attended design meetings with the client to provide technical input.

Ad hoc and on-demand communication: Individual developers and designers could communicate whenever there was a need for it. Communication could also take place via email or through phone calls, and occasional face-to-face sessions were held between designers and developers to work through proposed approaches. This communication had no prescribed structure.

5.2.2 Challenges during Engineering

Although developers and designers had good opportunities to communicate regularly, both teams agreed that communication challenges and inter-related challenges regarding the level of detail in upfront design remained.

Communication between Developers and UX Designers

Communication is a broad topic, but here, four main questions capture the challenge.

1. What information needs to be communicated? UX designers and developers did not always realise that there was a need for communication. There was not enough mutual awareness of each others' activities, leading to mistaken expectations that caused frustration. For example, this led to the independent production of two incompatible solutions for the same feature: one from the developers and one from the designers. The designers did not know that the developers had developed a solution, and the developers did not know that the designers were designing the feature without knowing the technical constraints. This mix-up caused frustration when developers saw the design solution for the feature (signed off by the client) that did not consider their technical solution.
2. How and when best to communicate information? This question points to the need for agreed ways of working. As an example, developers sometimes needed to tweak designs after they had been handed over by the designers. How and when should this be communicated to designers: when the need for a change is first realised? once a proven alternative has been implemented and tested? or somewhere in between? Developers may not expect that an early notification will be helpful. From a designers' perspective a continuous feedback process on the designs may be attractive in order to maintain a coherent design and to inform future decisions about the design, but it may also cause a high level of interruptions as a design route may take a while to develop.
3. How to keep communication channels open? Although various channels of communication were set up, issues sometimes took longer than expected to be resolved, and this held up development work. This was particularly true when developers had queries about the designs. Some developers felt inhibited about phoning a designer to discuss the issue. Instead they used more indirect channels like email, or tried to resolve the issue within the development team for the sake of speed.

4. How best to keep the design implementation visible? Limited visibility of the design implementation posed challenges for designers. This happened because the design implementation was not regularly reviewed by the designers. In our case study, although designers had the opportunity to review the design implementation, formal design reviews were only planned at the end of each increment for budgetary reasons, and this proved to be too infrequent to catch all the changes. More formal or more frequently-organised reviews would have raised design implementation visibility.

Level of Detail in Upfront Design: Sometimes Less Is More

From the developers' perspective the initial designs were unnecessarily detailed. They gave five main reasons for this.

1. Prioritisation and de-scoping can lead to a waste of pixel perfect designs.
2. Some issues with the design will only be found once implementation starts.
3. Pixel perfect designs may increase resistance towards making design changes.
4. It is better to focus on functionality first and design as you go along because when developing new functionality, "any visual work ... is a distraction."
5. Quality of designs can benefit from early input by developers with knowledge of design guidelines for the target platform.

5.2.3 Mitigations during Engineering

During the observation phase, LShift started to resolve the challenges by: introducing new roles and involving designers in development and vice versa. They also changed from a design-led to a development-led approach.

Introducing New Roles. Two standard DSDM roles were modified to help overcome challenges: the BA (Business Analyst) as communication bridge, and a Project Manager (PM) with experience of UX design and technical development.

The role was already introduced in Feasibility and Foundations. During the Engineering phase, this role developed into both a "communication bridge" between the developers and the client and between the developers and the designers to explicitly address the communication gap between them. This BA role was staffed by a senior developer, able to manage the discovery and communication of requirements, and to provide direct feedback on the technical feasibility of design ideas coming out of the meetings between designers and client. The BA also provided high-level requirements for the designers at the beginning of their sprints. Designers and developers perceived that the role improved communication.

A new person with experience of both technical projects and UX design was added to the team in order to take a classic DSDM Project Manager role. This was a departure from the company's usual model of employing a lead developer in the PM role, largely as a result of the size of the project and the amount of communication overhead around the design work. Doing so resulted in extra support for the extended BA role (described above), but also made sure that the designers' point of view was represented in the technical team.

Involving Developers in Design and Designers in Development. Visibility and transparency of the work by both parties was increased by:

- The designer attending daily stand-ups. Previously, although a designer attended stand-ups, different designers attended and they were often not the designer currently doing the work. Having consistency in attendance was perceived as positive, improved the communication and resulted in a quicker feedback loop.
- Providing access for designers and developers to all the stories being worked on at any one time.
- Releasing the implemented design to the designers once a week has led to iterative feedback coming from the designers to the developers.

From Design-Led to Development-Led Approach. Although the two mitigations discussed above improved the integration, a subsequent change of project requirements triggered a change of emphasis between development work and design work. Whereas previously a design-led approach was in play, a development-led approach was now needed. Technical spikes (prototypes) were developed and then shared with designers for their input. In more detail, this process involved:

1. Designers provided wireframes (not pixel-perfect designs) for the new functionality and these drove the conversation with the client and developers
2. Developers produced technical spikes (a ‘walking skeleton’) which cover basic functionality to complete a transaction or user journey to test a technical solution
3. Once the basic functionality was developed and agreed, the solution design was honed by the designers

This approach was perceived as useful since the design and the solution evolved together. Part of the reason for this switch of emphasis is natural as a result of the product concept maturing. There is a heavy emphasis on UX while the first requirements are becoming crystallised into new features, then as design patterns become clear, the development-led approach starts to become more prominent.

6 Discussion

This section returns to the research question posed above: “What challenges are faced by a company trying to integrate UX design and DSDM agile development?” In the next section 6.1, we discuss that the challenges found in a DSDM project are similar to the challenges in other agile teams. However, DSDM provides a different structure than other agile methods to address these challenges. We discuss this in section 6.2 and 6.3 focusing on the questions “How does the DSDM framework support this integration?” and “What implications do the answers to these questions raise for DSDM and other agile methods?”

In addition to the findings presented in this paper, a practitioners’ perspective on the lessons learned from this study can be found in [23].

6.1 Key Challenges in UX Design and DSDM Integration

The challenges faced by our DSDM organization find resonances in existing literature, and so are not unique to this method, and some existing mitigation strategies have proved useful in this context.

Communication between Developers and UX Designers. The role of face-to-face communication between developers and designers is stressed by Isomursu et al. [24].

Several publications suggest involving developers and designers in each other's process. For example, Budwig et al. [25] describes an approach in which the developers conduct design work by creating paper mock-ups, presenting them to the customer and then feeding back to the usability engineers. Designers can also be more closely involved in the development process, e.g. in the sprint planning meeting [24] or the stand-up meeting [17]. In our case study company, a designer did regularly attend the stand-ups, but it was not always the designer actively working on the project. Communication improved when the right designer attended.

Design collaboration can also be encouraged through, e.g., a design studio [26, 27] in which developers, stakeholders and designers produce design sketches, present them and critique them in order to find the best solution. The aim is to develop technically-feasible designs, and to promote a shared understanding, shared ownership of the design solutions and team communication.

Level of Precision in Upfront Design: This challenge is also faced by others. The level of detail required depends on the communication process between designers and developers, but the main message is "just enough". There is little guidance on exactly how much is "just enough", and reliance often falls back onto frequent communication. However, in our case study, developers suggested five main reasons why "less is more" when it comes to design documentation ready for the start of developer involvement.

Larry Constantine's classification of outputs as "deliverables" versus "consumables" provides a useful perspective [19]. Deliverables need to be finished rather than modifiable. On the whole, designs are deliverables for designers and consumables for developers.

6.2 How DSDM Supports Integration

In the case study organization, the roles of the extended Business Analyst and the hybrid Project Manager were seen as key to overcoming the challenges. The DSDM framework focuses on roles and phases, and hence does provide some support in this area. However, as there is no explicit mention of UX design in DSDM and no UX designer role, teams have to work out their own approach to managing this issue.

DSDM's approach to roles is more detailed than that of other agile methods, and this enables project teams to explicitly identify and manage team members with different specialisms and different levels of responsibility. The roles involved during Foundations are primarily the higher level 'Project' level roles, such as the Business

Sponsor, Business Visionary and Project Manager. During Exploration and Engineering responsibility moves over to the lower level ‘Solution Development’ team, which contains roles such as the Business Analyst, the Business Ambassador, the Solution Developer and the Solution Tester.

In this case study two DSDM roles were adapted, one higher ‘Project’ level role, the Project Manager; and one lower ‘Solution Development’ role, the Business Analyst. In DSDM the Project Manager is responsible for business and technical delivery, high-level management, the outline delivery plan and resourcing specialist roles. The introduction of a Project Manager with a background in technical and UX management meant that the project level decision maker was sensitive to the team’s UX challenges and was able to introduce new ways of working to improve the situation. The role of the Business Analyst in DSDM is to facilitate communication between the business and technical participants and to support Business Ambassadors and Advisors in thinking through requirements details. The extension of this role in the case study facilitated regular communication between the two teams.

These role adaptations were effective for LShift and could be attributed to the iterative research process with regular feedback points we initiated [21]. However, they were implemented fairly late on in the project and have yet to be fully evaluated. Others have also suggested introducing new roles to support team integration, e.g. Kollman et al have proposed a UX satellite [16]. An outcome from this work is a recommendation to the DSDM Consortium to include some explicit mention of UX design in the DSDM documentation along with some best practice guidelines that would help those using the method.

6.3 Implications for DSDM and Other Agile Methods

Three aspects of the findings here have implications for DSDM and other agile methods. Firstly, although UX design is not directly mentioned in the framework’s description, DSDM does provide some support for the integration, but it could do more. Specifically, having standard roles that can be extended to address UX communication issues as discussed in the previous section is a useful starting point. However the roles are not configurable in DSDM at present, but maybe this would be appropriate.

Secondly, five reasons for reducing the amount of up-front design were identified by the developers at LShift. Explaining these to designers at the beginning of the project may lead to less resistance from them to minimize wasted resources. This may be useful for all agile methods, including DSDM.

Finally, the shift between design-led and development-led project phases is worth highlighting. In our case, the teams could not have started the project in a developer-led fashion because the product brand and image needed to be established first. However, once enough of the design had been developed and the common vision was established, the developers were able to tweak existing designs in response to evolving requirements. This difference has not been highlighted before in any agile context, and could be better supported in both DSDM and other methods.

7 Conclusions

This paper focuses on the use of DSDM, but regardless of which agile method is used, the integration of UX design and agile development is a challenge. This paper has highlighted challenges and practical mitigations from which both DSDM and other agile methods may learn.

DSDM's Feasibility and Foundations phases offered a good opportunity to identify and mitigate challenges in communication between different expertises. The sooner these challenges are mitigated, the fewer challenges will seep through to Engineering. Challenges that arose during these phases were addressed using extended versions of DSDM's standard pre-defined roles. DSDM might therefore be enhanced by including guidance and support for such extensions, the exact details of which are likely to depend on the specific context. Explicitly including roles to support communication such as the communication bridge and the hybrid project manager may enhance other agile methods too. Other agile methods may also learn from the challenges that arose during the Engineering phase, where most agile methods focus.

In this case study, UX designers preferred to produce 'pixel-perfect' designs early on. This concern is common to many agile methods, but the reasons are not so often articulated. Here we have highlighted five reasons for these concerns, and these are applicable in all agile methods.

The nature of any agile project may change from design-led to development-led. In this case study, the importance of the UX role within the team reduced as soon as the principles were set and the development team was operating well. This may change the power balance in the project and hence the dynamics that create the challenges in the first place. All agile teams need to be aware of these differences. This change from design-led to development-led has not previously been discussed in the literature and we suggest conducting further studies to investigate how the power balance between designers and developers evolves during the course of a project.

References

1. Abelein, U., Sharp, H., Paech, B.: Does Involving Users in Software Development Really Influence System Success? *IEEE Software* 30(6), 17–23 (2013)
2. Constantine, L.L.: Process agility and software usability: Toward lightweight usage-centered design. *Information Age* 8(8), 1–10 (2002)
3. Sharp, H., Robinson, H.M., Segal, J.: eXtreme Programming and User-Centred Design: friend or foe?. In: *Proceedings of HCI 2004*, 2nd vol. (September 2004)
4. Chamberlain, S., Sharp, H., Maiden, N.A.M.: Towards a Framework for Integrating Agile Development and User-Centred Design. In: Abrahamsson, P., Marchesi, M., Succi, G. (eds.) *XP 2006*. LNCS, vol. 4044, pp. 143–153. Springer, Heidelberg (2006)
5. Garrett, J.J.: *The Elements of User Experience: User-Centered Design for the Web*. New Riders Publishers, Indianapolis (2002)
6. Dingsoyr, T., Dyba, T., Abrahamsson, P.: A preliminary roadmap for empirical research on agile software development. In: *Proceedings of AGILE 2008*, pp. 83–94. IEEE (2008)
7. Abrahamsson, P., Warsta, J., Siponen, M.T., Ronkainen, J.: New directions on agile methods: A comparative analysis. In: *Proceedings of ICSE 2003*, Portland, Oregon, USA (May 2003)

8. Fowler, M., Highsmith, J.: The agile manifesto. *Software Development* 9(8), 28–35 (2001)
9. DSDM Consortium (2013), <http://www.dsdm.org/dig-deeper>
10. Ferreira, J., Sharp, H., Robinson, H.M.: User Experience Design and Agile Development: Managing cooperation through articulation work. *Software Practice and Experience* 41(9), 963–974 (2011)
11. Ferreira, J., Sharp, H., Robinson, H.M.: Agile Development and User Experience Design Integration as an Ongoing Achievement in Practice. In: *Agile Conference (AGILE 2012)*, pp. 11–20 (2012)
12. Haikara, J.: Usability in agile software development: Extending the interaction design process with personas approach. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) *XP 2007. LNCS*, vol. 4536, pp. 153–156. Springer, Heidelberg (2007)
13. Broschinsky, D., Baker, L.: Using Persona with XP at LANDesk Software, an Avocent Company. In: *Proceedings AGILE 2008*, pp. 543–548 (2008)
14. Kane, D.: Finding a place for discount usability engineering in agile development: throwing down the gauntlet. In: *Proceedings of ADC*, pp. 40–46 (2003)
15. Cho, L.: Adopting an agile culture. In: *AGILE 2009*, pp. 19–25. IEEE Computer Society, Los Alamitos (2009)
16. Kollmann, J., Sharp, H., Blandford, A.: The importance of identity and vision to user experience designers on agile projects. In: *Proceedings of the 2009 AGILE Conference, Chicago, IL, USA*, pp. 11–18. IEEE Computer Society (August 2009)
17. Sy, D.: Adapting usability investigations for agile user-centered design. *Journal of Usability Studies* 2(3) (2007)
18. Sy, D., Miller, L.: Optimizing Agile user-centered design. In: *Proceedings of CHI 2008 Extended Abstracts on Human Factors in Computing*, pp. 3897–3900. ACM, New York (2008)
19. Patton, J.: (2013), http://agileproductdesign.com/blog/emerging_best_agile_ux_practice.html (accessed September 6, 2013)
20. Nielsen and Norman group report (2013), <http://www.nngroup.com/reports/agile-development-user-experience/>
21. Gregory, P., Plonka, L., Sharp, H., Taylor, K.: Bridging the Gap Between Research and Practice: The Agile Research Network. In: *Proceedings of European Conference on Research Methodology for Business and Management Studies (June 2014)*
22. Avison, D.E., Lau, F., Myers, D., Nielsen, P.A.: Action research. *Commun. ACM* 42(1), 94–97 (1999)
23. The Agile Research Network in conjunction with LShift Ltd: Integrating UX design into a DSDM project: challenges, working practices and lessons learned (2013), <http://agileresearchnetwork.org/ux-white-paper/>
24. Isomursu, M., Sirotkin, A., Voltti, P., Halonen, M.: User Experience Design Goes Agile in Lean Transformation – A Case Study. In: *AGILE 2012*, pp. 1–10 (2012)
25. Budwig, M., Jeong, S., Kelkar, K.: When user experience met agile: A case study. In: *Proceedings of CHI 2009 Extended Abstracts on Human Factors in Computing*, pp. 3075–3084. ACM, New York (2009)
26. Gothelf, J.: *Lean UX Applying Lean Principles to Improve User Experience*. O’Reilly Media (February 2013)
27. Ungar, J., White, J.: Agile user centered design: Enter the design studio - a case study. In: *CHI 2008 Extended Abstracts on Human Factors in Computing Systems*, pp. 2167–2178. ACM, New York (2008)