Integrating UX design into a DSDM project

challenges, working practices and lessons learned

Date October 31, 2013

Author(s) The Agile Research Network in conjunction with LShift Ltd.
Helen Sharp and Laura Plonka (The Open University, UK)
Peggy Gregory and Katie Taylor (University of Central Lancashire, UK)

Agile Research Network
Bridging the gap between research and industry

DSDM®
CONSORTiUM

1 Summary

Integrating user experience (UX) design into agile working is a challenge faced by many agile teams, not just those using DSDM. Although the work of UX designers and developers complements each other, the different goals, processes and working practices of developers and UX designers pose challenges in practice.

This white paper presents a chapter from the story of one organization's journey to integrate UX design into the DSDM framework. It describes the difficulties they faced, working practices adopted and the lessons learned from their experiences of integrating UX designers into their DSDM agile process.

LShift is a hi-tech software development company that works across a broad range of industries, languages and platforms. They are probably best known for creating RabbitMQ. They faced four main challenges while integrating UX design into the DSDM framework:

1 **Communication between developers and UX designers:** what is the relevant information that needs to be communicated, how to best communicate it, how to keep communication channels open and how to keep the emerging design implementation visible for feedback. Difficulties in these areas can cause frustration, problems with the technical feasibility of design solutions and mistaken expectations by the client.
2 **Level of precision in upfront design:** developers suggested five main reasons why "less is more" when it comes to design documentation ready for the start of developer involvement: prioritisation and de-scoping can lead to a waste of pixel perfect designs; some design issues will only be found once you start implementing; pixel perfect designs may increase resistance to making design changes; it is better to focus on functionality first and design as you go along; quality of designs can benefit from early input by developers.
3 **Design documentation:** the amount and detail of documentation needs to be discussed early on so that it meets both developers' and designers' requirements.
4 **User testing:** user testing can be a challenge in a product development setting if the business does not have customers yet. This can be addressed using personas and user representatives.

Several suggestions that would be compatible with LShift's situation and would ameliorate these difficulties, were identified from existing literature and discussed with LShift management. Some of these have been adopted.

Two roles in particular are relevant: an extended Business Analyst (BA) role and a hybrid Project Manager (PM). Communication problems arising between designers and developers can be overcome through a technically knowledgeable BA who acts as a bridge between the two disciplines, supported by a PM with both technical and design experience.

Visibility and transparency of the work by both designers and developers was increased by involving developers in design and designers in development: the same designer attending all daily stand-ups; all designers and developers having access to all current stories; and releasing the implemented design to designers once a week.

In this case study, subject matter experts delivered each area of the system, so there was no shortage of expertise. The main surprise was how quickly the shared understanding developed in the Feasibility and Foundations stage was lost once the teams started development.

2 Why UX and DSDM?

Producing high quality software that considers the needs of the users requires different professions to collaborate. While developers focus mainly on developing the code, user experience (UX) designers aim to provide the best possible user experience. Although the work of UX designers and developers complements each other and both aim to produce the best possible product for the customer, in practice designers and developers face challenges when working together.

Developers and UX designers have roots in different disciplines. They follow different processes, have different perspectives on software development and different working cultures. These differences present difficulties when integrating UX design into software development. One of the default assumptions in DSDM and agile methods in general is that UX designers and technical developers cohabit the same physical space, yet this is often not the case. For example, in small organisations the appropriate resources may not be retained in-house and in the bigger organisations the retained resources may not be in the same office or even the same continent.

This paper considers how UX design integrates with the DSDM method and presents some challenges faced by one DSDM Consortium member when integrating UX design into a DSDM project, together with the practices they adopted to overcome those complications. The case study was conducted by the Agile Research Network[1] with a high-tech software development company which has a core expertise in software delivery and subscribes to the DSDM method.

DSDM is an end-to-end framework for agile project management and delivery. The first version was published in 1995, and it has been developed through several versions since, with the most recent being the Agile Project Framework. The underlying philosophy is that projects must be aligned to strategic goals and focus on early delivery of real benefits to the business.

The DSDM framework covers the full project lifecycle including roles, process, practices, and products. Phases include Pre-project, Feasibility, Foundations, Exploration, Engineering, Deployment, and Post-project. It's highly configurable to accommodate a range of project types and size making it compatible with a variety of governance and programme office structures. The key practices used throughout the lifecycle are iterative development, MoSCoW prioritisation, Timeboxing, Modelling and Facilitated workshops. Additionally, DSDM provides a set of roles that ensures teams contain the right mix of representatives from the business, solution development, and management and process.

LShift chose to use DSDM to manage this project as the project was being commissioned by a largely non-technical client, who required transparency and predictability of delivery, so the classic light-touch agile approach would not provide enough management or visibility over the rate of progress. DSDM provides a mature set of tools and processes that help communicate with clients new to the agile process.

This white paper represents one chapter from the story of our case study company's journey to integrate UX design into the DSDM framework. It is told from a viewpoint outside the project and the companies involved, i.e. the ARN researchers. The

[1] The Agile Research network (DSDM-Agile-Research@open.ac.uk) is funded by the DSDM Consortium Board. The model operated by the network is that DSDM members propose the challenge they'd like to investigate, and then work closely with the research team to understand the causes and consequences of the challenge and to identify alternative ways of working.

chapter begins during the engineering phase of the company's latest design-oriented project. At this point the team were evolving their practices to address on-going challenges with UX design integration, and were looking ahead in order to anticipate the obstacles they might face in the future. ARN's role was to help identify their current difficulties and suggest practices that might help to mitigate them. As the case study company had been working with UX designers for some time, they had already faced and overcome some challenges. We begin with some background prior to ARN's engagement (Section 3), then describe the difficulties they still faced (Section 4) and suggestions for improvement from the literature (Section 5). Section 6 outlines what changes were implemented, and section 7 concludes the white paper.

Integrating UX design into agile working is a challenge faced by many agile teams, not just those using DSDM!

3 The story so far

The case study company does not employ designers themselves. Instead their UX design work is completed by a separate UX design agency that has experience of agile working but does not particularly subscribe to the DSDM method. The agency is a separate commercial entity and located in a separate building.

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.

In the initial stages of the project, the team had faced some obstacles created by the way designers and developers work (Section 3.1), and a new role had been introduced to overcome some of the concerns (Section 3.2). The need for an overall improvement in the communication between designers and developers had been recognised but not entirely addressed (Section 3.3).

Overview LShift
LShift is a high-tech software development company that works across a broad range of industries, languages and platforms. They have tried many flavours of agile, still actively using a few and have been a member of the DSDM consortium since 2002. They are probably best known for creating RabbitMQ.

Working with UX
A key feature of LShift work is the focus on collaboration with experts, such as UX designers. Due to the wide variety of projects LShift manages, they will usually collaborate with additional teams to produce projects, be they partner agencies or client-owned teams.

3.1 Different ways of working

Developers and UX designers have different ways of working and different perspectives on software development, and this can present frustrations in working together.

The case study company experienced the following specific differences:

Different perspectives and goals: Designers view their designs as their deliverable product while developers view them as consumables. For developers, working software is the deliverable.

Different processes: Designers prefer to develop complete UX designs upfront to ensure a coherent design makes sense throughout the application (i.e. BDUF, big design upfront), while agile developers prefer an iterative way of working.

Different commercial pressures: Trying to run agile teams as efficiently as possible involves trying to avoid blockages and duplication. As a result it's tempting to run in a staggered parallel fashion so that each team can focus on its area of specialization.

Different skills and knowledge: Designers focus on developing designs and associated documentation that satisfy the customer and that the customer will accept. Developers, on the other hand, focus on presenting the client with working software. Developers deal with technical constraints all the time while designers are not necessarily aware of the technical constraints under which their design solution needs to work. Similarly, designers work with design concepts which developers are not necessarily aware of.

3.2 Integration of UX and DSDM during Feasibility and Foundations

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

At the beginning of the Foundations phase, developers and designers were working mainly independently. 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 set their expectations, but they posed a challenge for the developers because some aspects of them were not technically feasible.

One of the main benefits of the design-led approach at an early stage is to focus on the potential features of the product unencumbered by practicalities of having to deliver them. Illustrating potential features using interface design mock-ups can also be a very useful tool for providing a shared understanding of what's to be built. However, the prioritisation of user functionality in agile projects meant that the implementation of some features was delayed to a later timebox than planned, and some features moved between increments.

Whether the functionality changed because of technical feasibility issues or prioritisation activity, the result was the same: the client did not receive the functionality expected according to the original timescale, and this was a challenge.

In order to mitigate this challenge, a developer was added to the UX team to work in a Business Analyst role, initially to specifically spend some time with the design team and help assess the feasibility of design proposals earlier in the process.

This role developed into a "communication bridge" between the developers and the client as well as between the developers and the UX agency. This is an extended version of DSDM's 'Business Analyst' (BA) role that incorporates some of the responsibilities of team leader, some of the project manager and with some elements of Scrum's Product Owner. The BA took ownership of requirements management and keeping the Prioritised Requirements List primed, so that stories were ready by the beginning of the Timebox. The Team Lead then managed the development team work and architectural decisions within the Timebox.

Lessons learned: Integrating UX into the early phases of a DSDM project

What issues need to be considered?

1 UX designers might not have the technical expertise to ensure technical feasibility of their designs. Developers won't necessarily know which elements of the design are feasible until stories are in development.

2 Developers and designers need to collaborate when developing designs and need to check for technical feasibility before design solutions are agreed with the client.

3 MoSCoW prioritisation may mean that some user stories will be developed later in the project than originally anticipated.

4 If the design is not modular the partial technical implementation of features can compromise the user experience.

5 The image of a product presented to the client at the beginning of the project will set their expectations. The nature of agile development means that this image may not be fulfilled. The client's expectations regarding the project need to be managed right from the start, and this requires both design and technical input.

6 Attending design workshops and client meetings is a time consuming task and additional technical resources will need to be included in the planning.

7 Designs are not only developed during the Foundations phase. Someone with technical expertise will be needed throughout the project to provide technical expertise to the UX designers. This will be discussed later in section 5.

What can be done about it?

Consider introducing an extended BA role to act as a communication bridge between designers and developers and to manage the client's expectations through the Foundations stage and beyond.

3.3 Integration of UX and DSDM during engineering

The developers' work was broken down into increments with three-week-long timeboxes. The team held daily stand-up meetings and ran planning meetings at the beginning of each timebox in which the work for the timebox was planned and estimated. The team worked with user stories that were organized in a shared project management tool.

The work of the UX designers was also timeboxed and the UX designers worked one sprint ahead of the development team (see figure 1). The designers organized their work during their timeboxes independently of the development team. However, the equivalent of half of one designer's time was allocated to support the development team during their timeboxes, to answer questions and to clarify design issues.

Designers and developers did not share an agile board as the designers preferred to manage their work independently and use a different tool for communicating over shared issues.

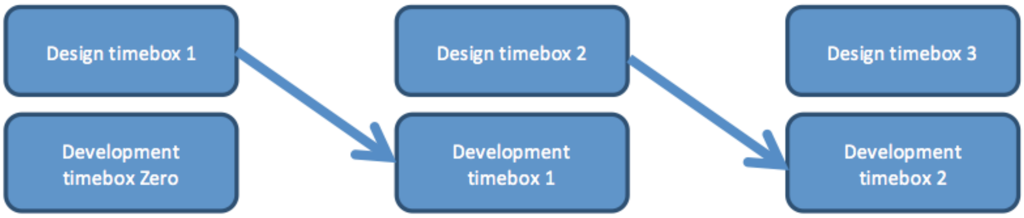

Figure 1: Designers and developers worked in timeboxes. Designers worked one sprint ahead of the developers.

The UX designers worked closely with the client working 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 via email. The developers received various documents. Usually designers handed over detailed, "pixel perfect"[2] designs 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, it was not feasible to run the classic DSDM whole-team workshops to facilitate communication, but the following communication-focused activities were in place.

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

The new 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. Specifically, the communication bridge worked directly with the client to drive the business requirements and also attended design meetings with the client to provide technical input.

Ad hoc and on-demand communication: Individual developers and designers could communicate on an ad hoc and on-demand basis whenever there was a need for communication. Communication could also take place via email or through phone calls. Occasional face-to-face sessions between designers and developers to work through proposed approaches. The ad hoc and on-demand communication had no prescribed structure.

4 The current state-of-play: Challenges remain

Both developers and designers work in an iterative and timeboxed approach, designers and developers have daily opportunities to communicate, developers and

[2] 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.

designers meet for daily stand-ups, a "communication bridge" between development team, designers and client has been introduced, so is a project set up like this still facing difficulties with the integration of UX?

The answer is yes. There is still room for improvement.

The remaining challenges identified by ARN revolve around four different but interrelated areas:

1 Communication between developers and UX designers
2 Level of precision in upfront design
3 Design documentation
4 User testing

4.1 Communication between developers and UX designers

Despite the designers and developers having the communication bridge role and the opportunity to communicate on an ad hoc and on-demand basis, both the development team and the design team agreed that the main challenge was still communication between the teams. This is a fairly broad area so we break it down into more specific questions below.

4.1.1 What information needs to be communicated?

Being given the opportunity to communicate is a good start but not enough. The opportunity to communicate does not address the question: *What information needs to be communicated?*

This is a crucial question because UX designers and developers do not always realise that there is a need for communication. For example, if designers are not aware that their design solution is not technically feasible why would they discuss the design solution with the developers? If a design that has been handed over to developers requires tweaking to accommodate technical factors, why would the developers consult the designers before implementing the tweaks?

In this case study, this situation 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 the developers were presented with the design solution for the feature (signed off by the client) that did not consider their technical solution.

In this example, there was not enough mutual awareness of each others' activities, and this led to mistaken expectations that caused frustration. Developers and UX designers had opportunities to communicate but were not aware of what information needed to be communicated.

4.1.2 How best to communicate information?

This question is a corollary to the previous one, and points to the need for agreed ways of working. As an example, developers sometimes need to tweak designs during implementation, i.e. after the designs have 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 an early notification to be helpful, but they may also not be aware of how their changes would impact the UX designers' work. 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. Such a process would be compatible with an agile way of working, but both developers and designers need to agree on how this would be achieved.

It is quite normal in team dynamics for the working practices to be in flux during the initial period of a project, while everyone gets used to their roles and personalities. With each member of the team focused on stabilising their own productivity, it can be too easy to forget to look over to see what else is going on.

4.1.3 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 just calling a designer to discuss the issue. Instead they used more indirect communication channels like email or tried to resolve the issue within the development team.

4.1.4 How best to keep the design implementation visible?

Limited visibility of the design implementation poses frustrations for designers. This can happen if the design implementation is not regularly reviewed by the designers. For example, if changes have been made to the designs during implementation, then designers need to review them before those parts of the software system are shown to the client.

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 visibility of the design implementations.

4.2 Level of detail in upfront design: Sometimes less is more

One of the common questions for UX design in agile software development is how much upfront design is needed? By 'upfront' we mean the amount of design work needed before developers are involved. From the designer's perspective the direction cannot be decided until each area of the product has been considered. It can therefore be a frustrating experience to be looking at a limited set of features at one time.

Getting the right level of detail in upfront designs is challenging. Developers and designers need to identify a) what needs to be provided for a developer to effectively implement the designs and b) how much upfront design is needed to ensure a coherent design. This requires UX designers and developers to come to a solution together.

In the context of this case study, developers and designers were working mostly independently and there was little visibility of each other's work. Developers were not involved in the design process and designers had limited visibility of the implementation of their designs. This lack of visibility and collaboration between developers and designers throughout different stages of the design process posed challenges.

From the developers' perspective the designs provided by the designers were very detailed. Five main reasons were given by developers as to why "less is more" when it comes to design documentation and the start of developer involvement.

1 **Prioritisation and de-scoping can lead to a waste of pixel perfect designs.** Due to on-going MoSCoW prioritisation and technical constraints, user stories may be deprioritised or changed throughout the engineering phase. This results in them not being implemented as envisioned. If designers spend time developing a pixel perfect design for all the stories up front, then this is a waste of resources and designs.

2 **Some issues with the design will only be found once you start implementing.** Developers said that questions or issues with the design are likely to be discovered only once developers start implementing the designs. Any such concerns will be reported back to the designers requiring designs to be retrospectively updated. The more detailed the design was to start with, the more costly the changes will be. Changing a pixel perfect design involves more re-work and is more time-consuming than changing a low fidelity prototype. An additional challenge occurs when the designs have already been signed off by the client.

3 **Pixel perfect designs may increase resistance towards making design changes.** Asking for changes in the designs late in the design process has another disadvantage from the developers' perspective: the resistance to change increases. If the designs have been signed off by the client, there is more resistance to change because this is what the client wants.

4 **No distraction, focus on the most important things first.** From a developer's viewpoint, it is better to focus on functionality first and design as you go along. Some developers said that they preferred to receive interactive wireframes instead of the pixel perfect designs because when developing new functionality, "the lack of visual styling can be very good because any visual work at that stage is a distraction." Note: designers would probably disagree with this statement, but this is a developer's comment.

5 **Quality of designs can benefit from early input by developers.** Sometimes user experience and user interactions are dependent on the platform and developers have relevant expert knowledge – including platform-specific user interaction standards and styles. In these cases, the design solution could benefit from developers and designers collaborating early on in the design process, not only to ensure technical feasibility but also to improve the user experience of the design. For example, in our case study some IOS developers had strong knowledge of the platform from both a coding and a UX point of view. This included understanding the user experience of touch screen devices and platform-specific HCI guidelines.

4.3 Design documentation

This challenge is related to the previous one. The question is how much design documentation is needed? In our case study, developers and designers had a different perspective on this.

Some developers said that designers initially produced “lots and lots” of documentation. Detailed documentation is hard to keep up to date, so after some design iterations it was not clear which parts of the documentation were up-to-date and which were not. One developer explained that he spent “lots of time going backwards and forwards with the design agency discussing things with them, and getting updated elements, and trying to get clarity on how things should work.”

From a designers’ perspective, detailed documentation was important to explain how each of the elements and interactions work. Designers said that if they don’t describe everything in detail, developers get back to them and ask them about it.

The design team were keen to keep a body of detailed documentation in pace with the work as it progressed through the timeboxes. This was felt to be necessary because the solution needed to be reviewed against the "proper designs" to judge implementation quality.

This was dropped later in the project once the solution had developed sufficiently for the quality criteria to be judged by the actual experience of using the solution.

Despite these different perspectives, the design documentation did not present a major challenge to the teams. When exchanging their perspectives in a retrospective, designers and developers realised that they had not given feedback about their concerns over detailed design documentation before.

4.4 User testing

Frequent feedback is one feature of agile projects, and one form of this is the customer demonstration which takes place at the end of each timebox. However this is not the same as user testing. In user testing, representative users are asked to use the product while performing realistic tasks, and their performance is captured for later analysis. So how does user testing fit into the DSDM lifecycle? Who is responsible for conducting user testing? And how systematic does user testing need to be? These questions were raised in our case study, and further characteristics of the setting generated additional challenges. In a product development environment such as this, user testing can be a challenge as the business does not have customers yet. The team tried to work around this limitation by various means:

- Personas. Archetype fictional users were created. These illustrated characters were designed to represent the different types of end user.
- User representatives. As Ambassador users were not available, these roles were filled day to day on the project by a couple of individuals.

At a later stage in the project when real users became available, active feedback was sought. The accrued design and build debt was dealt with as a by-product of the process.

5 What does the literature suggest?

This section focuses on ideas from the research and practitioner literature that address the issues faced by our case study company. These suggestions are tailored to this specific context. Other references in the next section point to the wider literature in this area.

Existing research acknowledges that integrating UX and agile can be challenging. Kollmann et al. [8] stresses the importance of UX designers and developers developing a **common vision and common ownership** together. Ferreira et al. [4] points out that on a day-to-day basis the integration of designers and developers is achieved through **mutual awareness, expectations about acceptable behaviour, negotiating progress and engaging with each other**.

5.1 Suggestions to improve communication and collaboration between developers and designers

The role of face-to-face communication between developers and designers is stressed by Isomursu et al. [7] who state that "tool support for collaboration does not succeed in solving the communication problems arising from inadequate opportunities for face-to-face communication."

The following suggestions are relevant for our case study context. The aim is to create opportunities for designers and developers to communicate and collaborate.

5.1.1 Involve developers in the design process and involve designers in the development process

Several publications suggest integrating designers and developers by involving them in each other's process.

Budwig et al. [3] describes an approach in which the **developers conduct design work** by, e.g. creating paper mock-ups, presenting them to the customer and then feeding back to the usability engineers.

On the other hand, designers can also be more closely involved in the development process. A good opportunity is the sprint (timebox) planning meeting [7] in which developers estimate and break down tasks. Having UX designers participate in the **sprint planning** allows developers and designers to define UX tasks, and developers can clarify questions about the design. Another opportunity is the **stand-up** meeting [12]. After these meetings, designers can report on various UX design activities including successes with design prototypes, contextual information about users' workflows but also on problems with design prototypes or bugs in working versions. Sy [12] also suggests capturing design issues as **story cards on a UX board** to increase their visibility.

5.1.2 Design collaboratively: Design studio [5], [13]

Design collaboration can be encouraged through, for example, a design studio. A **design studio** is a one-day workshop that includes developers, stakeholders and designers and is led by UX designers. During the workshop everyone produces design sketches, presents them and critiques them in order to find the best solution. Aside from developing designs the aim is to promote a shared understanding, shared ownership of the design solutions and team communication. It also provides the opportunity to rapidly explore of design alternatives and to discuss the technical feasibility of the design solutions early on as each participant will contribute according to their expertise.

5.1.3 Introducing specific roles into the process

The integration of design and development can be also facilitated by introducing specific roles that focus on the integration.

In the context of Scrum, Budwig et al. [3] and Singh [11] both describe adding a **UX product owner** to the team to ensure that UX aspects are considered by the development team.

In Budwig et al.'s [3] case study, both the development team and the UX team were organized as scrum teams each with its own product owner. The UX product owner collaborated closely with the development product owner, wrote UX stories and coordinated UX deliverables and roadmaps with the development teams. According to Budwig et al. this "ensured that the UX deliverables were considered in all stories".

Singh [11] has a similar perspective but focuses on adding a UX product owner to a software development team to ensure that the overarching user vision is taken into account. According to Singh [11], traditional product owners often choose small tasks "they are confident could be completed within a single sprint". This perspective might lead to future challenges including poor usability.

Another strategy reported by Kollmann et al. [8] is having a UX person as a **satellite** on the development team. The UX satellite focuses on the relationship with the developers and is supported by the rest of the UX team who conduct usability testing or produce prototypes or screen flows.

5.1.4 Visual representations as reminders

One of the major challenges is to identify what is the relevant knowledge that needs to be shared. One strategy is to have visual representations of "what is going on". These representations can prompt conversations and act as reminders.

Kollmann et al. [8] describes the idea of a "**Question Board**". The "Question Board" was started by developers who had design-related questions but it is used by both designers and developers. Kollmann et al. [8] explains 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 and viewpoints.

Another strategy to make design work more visible and keep developers informed about the on-going design work is to provide **posters of personas** in the development team area [2]. Personas can act as reminders to developers about who they are developing for. Broschinsky [2] points out that it is important for the user research results leading to the personas to be explained upfront to provide an understanding of how the personas emerged.

5.2 Suggestions focusing on the level of detail in upfront design

The level of detail required depends on the communication process between designers and developers, but the main message is "just enough". However there is less guidance on exactly how much is "just enough", and reliance often falls back onto frequent communication. Larry Constantine's classification of outputs as "**deliverables**" versus "**consumables**" provides a useful perspective [10]. Deliverables need to be finished rather than modifiable. On the whole, designs are deliverables for designers and consumables for developers.

5.3 Suggestions focusing on documentation

The website Agile Modeling by Scott Ambler [1] provides a detailed description of "best practices" for agile documentation. These practices support writing "good enough" documentation in an efficient way. The focus of the suggested practices is to determine whether and how much documentation is needed. The practices do not suggest what exactly needs to be documented but they rather propose questions such as "what is the purpose of the documentation", "who is the customer of the documentation" and "when should documents be updated" to determine what and how much is needed. Details about each practice can be found on his website.

5.4 Suggestions for user testing

Issues commonly faced when trying to integrate user testing into agile projects include frequency of user testing [5], [6] and how to make the best use of user time and to elicit good user feedback [9]. The DSDM method itself suggests regular testing with stakeholders, including end-to-end usability testing, but this also requires available users. However given the constraints in this case study due to the lack of users, incorporating any standard approaches to user testing would be impractical.

Personas are commonly used alongside agile processes, not just in the early phases of a project. While this does not replace user testing, it does help to keep developers focused on specific user types during development [2].

6 What did they do?

Communication and collaboration was identified as the key challenge to be addressed. If this could be improved, then other challenges would be reduced. The approaches identified as being most appropriate for this setting were: introducing new roles, involving designers in development and vice versa, and designing collaboratively. The last of these has yet to be implemented, but the potential of a co-creation workshop or a design studio is positive.

6.1 Introducing new roles

Two standard DSDM roles were modified to help overcome communication and collaboration challenges: the BA as communication bridge, and a PM with experience of UX design and technical development. In addition, a UX satellite role will be introduced again when the level of design work warrants it.

6.1.1 The BA as Communication Bridge

The communication bridge role supports explicitly the communication gap between developers and designers. This BA role was staffed by a senior developer, able to manage the discovery and communication of requirements, but also to provide direct feedback on the technical feasibility of design ideas coming out of the meetings between designers and client, another is to provide high level requirements for the designers at the beginning of their sprints.

- **The communication bridge role and requirements.** When talking about this role, the designers focus on the task of elicitation and providing requirements. Designers describe that prior to the introduction of this role the requirements had not been communicated on time. This had resulted in a very uncertain design process and the need for re-work because the designers had tried to write the stories themselves in the absence of requirements. Designers perceive that the communication bridge role improved the on-time communication of high level requirements.
- **The communication bridge role and technical feasibility of the design.** Developers perceived that introducing the communication bridge role improved the technical feasibility of the design solutions. However, designs might impact very specific areas of the technical solution. In this case, additional technical input by the subject matter experts at an early stage of the design would be beneficial to understand the full breadth of how the design might impact the technical solution.

6.1.2 A new Project Manager (PM)

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 has resulted in extra support for the extended BA role, but also making sure that the designers' point of view is represented in the technical team.

6.2 Involving developers in design and designers in development

Visibility and transparency of the work by both parties was increased by:

- The same 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. This resulted in the need to spend time updating the new designer about what had happened before. Having consistency in attendance was perceived as positive, improved the communication and resulted in a quicker feedback loop.
- Providing access to all the current stories for all designers and developers.
- Releasing the implemented design to the designers once a week, which has led to iterative feedback coming from the designers to the developers.

6.3 Where are they now

During the most recent increment the working pattern has seen a change in emphasis between design work and development work. Whereas previously a design-led approach was in play, project requirements have meant that a technical-led approach is now needed. Technical spikes (prototypes) are developed and then shared with designers for their input. In more detail, this process involves:

- Designers provide wireframes (not pixel-perfect designs) for the new functionality and these drive the conversation with the client and developers.
- Developers do technical spikes (sometimes called walking skeletons) which cover the basic functionality to complete a transaction or user journey to test a technical solution.
- Once the basic functionality is developed and agreed, the solution design is honed by the designers.

This approach is perceived as useful since the design and the solution evolve together, whereas if a detailed design had been developed upfront, then more changes would be required after creating the technical spikes. It is less clear whether this approach could have been taken from the beginning of the project, since a good understanding of the overall product design (for both developers and designers) has evolved over time.

7 Is this a common challenge? Yes!

It's not only DSDM projects that face challenges relating to the integration of UX design into agile projects. There have been several experience reports and academic studies published in this area over the last 10 years, and it is a common discussion topic among communities of UX practitioners. In fact, DSDM is in a strong position to address this common problem because it has clearly defined roles that can take on the responsibility for UX design.

In the case study presented here, there were subject matter experts delivering each area of the system, so no shortage of expertise. The main surprise to the team was how quickly the shared understanding that was developed through Feasibility and Foundations was lost once the teams put their heads down into the development process.

Other organisations have found it necessary to have BAs or Product Owners representing each team and negotiating the product roadmap between them. The resolution this team found was to place a single technically knowledgeable BA into a bridging role between the two disciplines.

The fact that two distinct companies were involved – one employing designers and one employing developers – did affect the challenges, but these difficulties are common within projects of different structures too. And there are other challenges – often described simply as 'communication problems', but this covers a myriad of possibilities and requires detailed investigation in each case.

If you are facing similar problems in this area, you may find it useful to read the sample of publications below, or to contact the Agile Research Network as a collaborator.

References

These references constitute only a small subset of the literature on agile UX.

[1] Scott Ambler. *Best Practices for Agile/Lean Documentation*. 2013. URL: `http://www.agilemodeling.com/essays/agileDocumentationBestPractices.htm`.

[2] David Broschinsky and Lisa Baker. "Using persona with XP at LANDesk software, an avocent company." In: *Agile, 2008. AGILE'08. Conference*. 2008, pp. 543–548.

[3] Michael Budwig, Soojin Jeong, and Kuldeep Kelkar. "When user experience met agile: a case study." In: *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. 2009, pp. 3075–3084.

[4] Jennifer Ferreira, Helen Sharp, and Hugh Robinson. "Agile Development and User Experience Design Integration as an Ongoing Achievement in Practice." In: *Agile Conference (AGILE), 2012*. 2012, pp. 11–20.

[5] Jeff Gothelf and Josh Seiden. *Lean UX: Applying Lean Principles to Improve User Experience*. O'Reilly, 2013.

[6] Zahid Hussain et al. "Integrating Extreme Programming and User-Centered Design." In: *PPIG'08: Proceedings of the 20th annual meeting of the Psychology of Programming Interest Group, Lancaster, UK*. 2008, pp. 107–113.

[7] Minna Isomursu et al. "User Experience Design Goes Agile in Lean Transformation–A Case Study." In: *Agile Conference (AGILE), 2012*. 2012, pp. 1–10.

[8] Johanna Kollmann, Helen Sharp, and Ann Blandford. "The importance of identity and vision to user experience designers on agile projects." In: *Agile Conference, 2009. AGILE'09*. 2009, pp. 11–18.

[9] Jennifer Jen McGinn and Ana Ramírez Chang. "RITE+ Krug: A Combination of Usability Test Methods for Agile Design." In: *Journal of Usability Studies* 8.3 (2013), pp. 61–68.

[10] Jeff Patton. *Twelve emerging best practices for adding UX work to Agile development*. 2013. URL: `http://agileproductdesign.com/blog/emerging_best_agile_ux_practice.html`.

[11] Mona Singh. "U-SCRUM: An agile methodology for promoting usability." In: *Agile, 2008. AGILE'08. Conference*. 2008, pp. 555–560.

[12] Desiree Sy. "Adapting usability investigations for agile user-centered design." In: *Journal of usability Studies* 2.3 (2007), pp. 112–132.

[13] Jim Ungar and Jeff White. "Agile user centered design: enter the design studio-a case study." In: *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. 2008, pp. 2167–2178.