This experience report covers three agile projects, each with a significant UX element. For each project the paper outlines:

- An overview of project challenges
- Observations from an agile UX perspective
- Improvements suggested by the team, on reflection of their experiences

The summary outlines tools and techniques used to address indentified challenges, taking into consideration observations and improvements.

Author: Matt Roadnight

August 2010

matt.roadnight@sprintagile.com

1 Introduction

This paper is a summary of some projects that I have had involvement with over the past few years, some good, some not so good, but always a learning experience. As more projects that include a User Experience (UX) element are adopting an agile approach I have experienced tensions surface with the way that UX specialists work and the demands that an agile project puts on them. This appears to be common as I have heard similar stories and challenges on other projects.

I have put together this experience report as a reference to highlight the benefits of addressing those tensions, but also some advice to those that are not willing to adapt UX practices on agile projects. These experiences highlight:

- Gaps that can appear between different specialists on a project, with a focus on UX
- The impact on projects where specialists don't adjust their approach to be more collaborative
- How projects can benefit from a shift in approach, focusing on UX specialists and artefacts

I have outlined three projects; each is presented with a context setting overview, some project observations focussing on agile and UX tensions and an outline of what the teams said they would do differently next time.

Project One a large product with multiple teams and up to five UX specialists

Project Two a smaller project that came to an initial sprint with little upfront UX artefacts defined

Project Three a project that had to work with a brand sign-off process that expected high fidelity artefacts

Each of these projects used Scrum as their agile approach. If you are not familiar with the process, roles and artefacts please refer to Reference [R1], references are provided on the last page.

In the summary I cover principles and techniques I have used to assist UX specialists on agile projects. These approaches utilise standard UX artefacts to ease tensions and help UX specialists work more effectively on agile projects.

2 Project One

2.1 Overview

This project team was made up of about 30 people that included Project Managers, User Experience Architects, Designers, Copywriters, Front End Developers, Component Developers, Data Specialists, Platform Architects, Testers, Business Analysts; talented specialists from all areas of web development to build a great online solution.

They chose to use Scrum as their project management framework and after doing some upfront research and getting development infrastructure in place during an initial sprint 0 (preparation sprint) they launched into one month sprints.

2.1.1 A good start

The UX specialists (User Experience Architects and Designers) began to develop artefacts to deliver a well thought out and detailed user experience. They had a number of workshops with a broad customer stakeholder base to validate their research and start to build and deliver the following artefacts:

Personas User Journeys Design Mocks ups HTML Mock ups Wireframes Style guide

These artefacts were the type of artefacts that the UX specialists had produced in the past for waterfall projects, they felt there was no need to change this. The work the UX team put together was first rate and the business stakeholders they collaborated with to produce these artefacts were impressed and happy with the results.

At the same time the development team (Front End Developers, Component Developers, Data Specialists, Platform Architects, Testers) pushed forward on the development delivering initial architecture and started to add basic features to the solution. Both UX and development were progressing in parallel, with UX looking to get the whole site designed, agreed and signed off with the stakeholders. Development was also busy addressing challenging early project issues.

2.1.2 Some cracks appearing?

This was all fine for the first few sprints, UX were well through most of the design and layouts, most of the site had been reflected in UX artefacts that were reviewed and agreed with the customer, however there were a few signs that all may not be well….

The daily scrums were being run for the whole project team, the UX specialists weren't getting any value from the daily scrums; they were wasting precious design time and were up against tight deadlines. The development team were really focussed and having some key challenges themselves, the daily scrums sometimes went over 15 minutes as the development team gave their updates and discussed issues. The UX specialists switched off in the daily scrums and eventually all agreed that they should stop participating.

The initial sprint reviews with the customer also had some challenges; the teams took on a lot of work every sprint in order to get enough done on the project, but were failing to complete all of the work sufficiently. The unfinished work each sprint wasn't going down well with the business customer and the quality of the output that was being demonstrated wasn't in line with the "pixel perfect designs" they had expected to get each sprint.

Over time the business customer lost faith in the monthly review sessions, so they were dropped. The team were also too busy for retrospectives, these too were dropped.

The project got to the stage when all of the UX artefacts were complete and agreed with the customer. Budget for UX specialists was squeezed, the development team continued to work on developing the functionality defined in the designs and wireframes. Most on the project were familiar with the overall solution, as the designs reflected their understanding; they were so busy on focussing on doing the work that they omitted to keep the product backlog up to date, focussing instead on the UX artefacts.

The team eventually delivered an excellent solution, with a first class feature set and user experience, but it was over budget and very late. Neither the team nor the business customer had the best delivery experience.

2.2 Observations

The UX specialists had run a long way ahead of the rest of the team in terms of UX design and there was little understanding of the effort required to implement that design. They understood their remit was to deliver a first class user experience; this was done with little collaboration across the whole project team.

The underlying reasons for this were many; however there are a number of observations that may help other projects avoid this situation.

2.2.1 The power of UX artefacts

Everyone loves visual images to support web site design, they convey a lot of information and give business users and stakeholders some comfort that the team have taken their ideas and interpreted them correctly. They also provide the opportunity to explore concepts, to agree designs and look and feel.

The fidelity of artefacts that UX specialists produce can set early expectations on the final solution. This isn't a perceived problem with a more waterfall approach. For an agile project however they may provide a false sense of accuracy or completeness that may not be known far in advance. This may hinder the ability of an agile project to support collaboration with the customer in order to meet budget and time restrictions.

This project had been set up as an agile project, therefore the initial definition of the scope was somewhat loose and open to interpretation. The stakeholders had agreed to a level of scope that met their budget. This however hadn't been clearly communicated to the team who were more closely aligned to delivering a first class experience, and duly went about working towards that vision.

2.2.2 Buy in to an agile approach

Many of the development specialists had been on agile projects previously or attended agile courses and shared experiences on agile projects with others. Many of the UX specialists and project management team had some exposure to agile, but didn't fully buy into an agile approach. Therefore there wasn't sufficient support for a more agile approach to delivery i.e. exploring other approaches to delivering UX artefacts or using the product backlog effectively to manage scope, supporting more effective cross team collaboration and adhere to basic Scrum rules.

The stakeholders also didn't have an appreciation of what was expected of them and failed to get engaged with the team to build the solution together. This separation was also exaggerated by the fact that there was an organisational boundary between the team and stakeholders, the stakeholders had a more traditional customer / supplier relationship in mind. Due to this position, after agreeing to initial designs across the whole solution, the stakeholders were not open to discussing the product backlog or scope when it became apparent that there was too much work left to deliver in the agreed time.

2.2.3 Product Backlog Housekeeping

Due to the lack of understanding of agile and the fidelity and richness of the UX artefacts the product backlog lost its importance, the team instead focussed on the UX artefacts. The product backlog lost its usefulness as a tool to manage scope and forward plan. The team focussed on artefacts that they had more affinity with on more traditional projects and used these to build the solution with, rather than search out answers to more loosely defined requirements in collaboration with the stakeholders.

Another side effect of not keeping the product backlog up to date was that the team could no longer give a forward view as to when they might complete the work, the backlog no longer accurately reflected what work they had completed and what needed still to be done. The remaining work on the product backlog was not well defined and included items that had already been built. This in turn affected the ability of the team to accurately estimate work to take on and complete in a sprint as it lacked definition and there were still questions with the solution outlined in the UX artefacts.

2.2.4 Collaboration

The UX team worked how they had always worked; creating artefacts that reflected the design of the whole solution, working closely with the customer, but not working closely enough with the rest of the project team. This was partially due to the fact that they were "too busy" but they also didn't buy into the need to collaborate with the rest of the team, they had always worked that way and no one on the team pushed the matter.

As the project progressed it became clear that the UX artefacts prescribed complex solutions that meant the development team needed to spend more effort than expected in order to meet expectations. Some of the designs needed further discussion due to idiosyncrasies that couldn't be implemented, or needed further clarification before they could be developed. The UX specialists had since moved on and were difficult to get hold of, so the remainder of the team had to iron out these issues with the stakeholders. The team had to resort to writing specifications to agree the finer details described in the UX artefacts.

As the UX specialists had since moved onto other projects the opportunity to get usability feedback on the software that had been built within the sprints had been lost.

2.2.5 Agile mindset

This isn't specifically a UX gap issue, but more of an issue that I observe with "challenged" agile projects. The team were going through the motions of applying the Scrum framework, without an appreciation for the underlying principles that need focus and appreciation in order to support collaborative working and effective use of an agile approach.

This requires energy, passion and commitment. It's easy to lapse into old waterfall based habits, but unless they are supported by appropriate waterfall artefacts they won't be as effective as they were previously. You have to live by agile principles and perceiver with what you know is right, possibly against some challenging opposition.

2.3 What they would change next time

Having spoken to the team there were a number of areas that they would improve in the future, some of the higher priority areas are outlined below.

2.3.1 Gain input cross functionally across the project

The team realised that the different specialist within the project didn't work effectively as a team, they would have benefitted greatly from a more joined up approach. Their principles for addressing this next time would be:

- Regular reviews of designs and working software within the team and not just the stakeholders
- Make sure everyone discusses functionality, understands the requirements and has more clearly defined joint deliverables
- Bring one of the end users into the team to work through issues with them
- Ensure that all specialism's are represented throughout the project
- Be on top of the product backlog daily and make it visible to the whole team

There was a general agreement that there was a lack of clarity on the requirements, there were too many assumptions and projects members needed to question and clarify more often.

Ideas for functionality should get put on the product backlog; there were too many suggestions that got designed in the UX artefacts without being made visible. The UX artefacts that were used to drive the delivery left many questions unanswered and were not backed up by clearly defined product backlog items.

Had the product backlog been used more effectively, it was acknowledged that it would have been a useful tool to manage scope and expectations.

2.3.2 Keeper of the Vision

Had the vision been more clearly articulated and the product backlog been kept up to date, the team would have stayed more focussed on cross functional tasks and the UX specialists would not have worked too far ahead of the rest of the team. Expectations set in discussions with the stakeholders could have been captured and the impact on the overall project identified.

2.3.3 Set expectations on delivery

The team consistently took too much work into sprints, failed to complete the work and finished sprints with a number of outstanding bugs that were not addressed. This was due to the pressure of having to hit deadlines, and poor backlog definition. Next time they would define and religiously stuck to their definition of "DONE" [G5] and not take too much work into sprints.

2.3.4 Education

There wasn't a common understanding of the process within the team or the stakeholders or what was expected of each role. Had there been more time spent at the start of the project educating everyone the team felt this would have made a big difference.

3 Project Two

3.1 Overview

This team were new to agile and when I met them they had just completed their first one month sprint, had got the basic infrastructure in place for a development and system test environment and had completed the installation and the basic configuration of a content management system (CMS).

During the first sprint they had run a story-writing workshop as outlined in Mike Cohn's book "User Stories Applied"[R2]. Adam was the main stakeholder, he held the budget, had gained buy in from the rest of the company and had selected a delivery partner. He had been thinking about the project for a while so had a good idea of the functionality they were looking for and had engaged a visual designer who had captured the look and feel in a number of visual concept Photoshop images.

The team were relatively small and from a consultancy comprised of a UX designer / developer, three developers, a tester and a scrum master. There were a number of business stakeholders from the client who were also involved with the team.

I met with the team just before they were going into their first sprint review meeting; I discussed their plans for their review, the retrospective and the planning meeting for the next sprint.

The planned agendas and outcomes for the Scrum meetings over the next few days weren't exactly "Scrum by the book", but the team seemed clear on what they wanted from the meetings and had already set expectations with the customer. I didn't want to undermine the team and alter what had already been agreed.

The team were planning to use the Scrum review and planning meetings to discuss the product backlog priority, review the concept design, run some group home page design sessions and derive user flows and pages to support the highest priority user stories on the product backlog. Planning tasks for the next sprint wasn't mentioned.

3.1.1 Review Meeting

The sprint review was well attended by stakeholders and end users from each of the departments. The review went really well, the team completed everything they committed to and it was the first time that the stakeholders and users had seen what the team was working on and they enjoyed using the new CMS, as it was an area of pain on the current site.

The review of the visual concepts went well, and there was a large amount of discussion of ideas from each of the end users.

The product backlog had been captured on cards from the story-writing workshop; there was no priority or structure to group all of the individual user stories. I suggested that we identify some "themes" [G2] to collect the stories under and started to generate a user story map [R3] and attempted to draw a line under each theme to understand where an acceptable level of functionality could be for an initial final release. Adam refused to prioritise even at this high level so we moved on and started to discuss some of the product backlog items in more detail.

3.1.2 Planning Meeting

As with the previous day, a large number of stakeholders and end users attended, with eight representatives from the customer attending along with all the team members. The review of the highest priority backlog items was intense with discussion; there had been little opportunity prior to this meeting to discuss how each of the stories would be implemented.

There were also few visual artefacts to reference, the artefacts that did exist were conceptual in nature and hadn't taken into account business or technical constraints or detailed business requirements. Little up front work had been put into how the conceptual designs would interact or understanding the flows between the different system states.

With so many people in the room the discussion was frantic, with many different views on what was required, how the system data should be organised or represented, and how the system should interact with the user.

A lack of visual reference for the discussion meant that many of the areas being discussed required everyone to visualise it for themselves, or draw up sketches on the white board. As there were so many different views and a broad set of opinions being discussed over a large functionality set it was difficult to get to a consensus or capture the conversation in enough detail to support effective development of a solution.

At the end of the sprint planning meeting the goal for the day hadn't been reached, only a small number of the stories had been discussed in any detail and there were still a number of open questions about those. The team didn't have enough time remaining to plan in detail; they did have story points [G1] on their backlog, so they took an equivalent number of stories in to this sprint, based on the last sprint's progress.

3.1.3 The second sprint

The first few days in the sprint were spent with the visual designer and UX designer / developer taking the concepts and the knowledge gained from the planning meeting along with the group session sketched home page designs to validate their thinking.

This meant they worked through white board sketches of key screen layouts and flows to cover the scope of the backlog items taken into this sprint.

The UX designer was also the front end developer so the step to take those sketches and ideas into actual working software didn't need any further artefacts within the sprint. It only required a reference back to the concept designs and digital photographs taken of the sketches captured on the whiteboards.

The front end development in this sprint focused on building a XHTML/CSS component library to generate greyscale wireframes, and linked together to form a prototype in the system test environment. The layout of the visual concepts and wireframes were based on a common grid structure (960 grid) - therefore easing the translation of visual design to prototype. Each of the themes defined on the product backlog were represented by components on the pages. In some cases the components imitated the intended functionality with JavaScript behaviours. As development evolved these place holders were developed with actual functionality.

As the end of first sprint approached it was clear that the team had taken on too much work, and hadn't been giving the stakeholders regular deployments of functionality to allow them to feedback.

The sprint review meeting was difficult as much of the promised backlog items had not been delivered and those that had been, took a long time to review as it was the first view that the stakeholders had for much of the functionality. The few that had been released to the stakeholders in the sprint had a large number of bugs outstanding.

3.1.4 Following sprints

A number of changes were introduced at the end of the second sprint to address some of the key observations; these are detailed in the next section. The future sprints ran their course, based on the progress reflected in the product backlog. The stakeholders collaborated to make some key decisions on functionality and release dates, and the team delivered a great site that the customer is very happy with.

3.2 Observations

The team had entered into the sprint planning meeting with little or no preparation up front for the stories on the product backlog or the supporting UX designs. From a Scrum perspective the result was a pretty ineffective planning meeting, resulting in a sprint that had far too much work to complete.

However with some focus on the product backlog and an open mind to trying a different approach the project got back on track.

There were a number of observations that were identified in the initial sprints. Some of these observations go towards answering common question for UX on agile projects, and steps taken to address those issues, these are outlined below.

3.2.1 How much work up front?

This project had a lot of forethought by the stakeholders prior to kicking off the "build" process and starting the sprints. This was reflected in tender and bid documents published by the customer and some Photoshop concept designs that reflected what the site could look like.

What had been discussed and agreed upon were some of the areas of "high cost of change". From a technical perspective the platform had been agreed (development environment, language, products etc) and from a UX perspective the overall look and feel had broad agreement and some initial concepts had been created (basic page layouts, and designs). These are areas that generally take longer to agree with stakeholders, but once agreed provide a general direction or "framework" to work within.

The level of detail required to support a sprint planning meeting was not sufficient, given the level of discussion in the room. You will know if you don't have enough information supporting your stories, you will experience a similar level of discussion and disagreement.

The front end developer in this instance had UX design skills, and provided the interaction design; the concepts were developed by a visual designer. Both of them worked together very closely, this was a key success factor. The artefacts for this team to turn concept designs to working software were based on an understanding supported by collaboratively developing low fidelity sketches and ongoing conversation. In some cases the bigger design ideas were also discussed with the customer first, but for most of the smaller decisions these ideas were validated and refined when the customer reviewed the working software.

There was little need for formal wireframes or detailed mock ups, ideas and concepts were communicated through discussion using the sketches and concept designs. Initially these sketches were developed within the sprint, but as the project progressed these were generated just ahead of the next sprint, to address the disagreement in the planning meetings. This approach has been used successfully by agile practitioners across the industry and has been referred to as getting backlog items "ready" [G4].

There was some revisiting of the overall front end development in later sprints. The application of style wasn't consistent throughout the site, and some of the user interface needed tidying up to align pixels, drop down behaviour and icon usage etc. This work was placed on the product backlog, planned and addressed in priority order.

The benefit of lower fidelity, faster generation of artefacts for the larger pieces of UX work (e.g. screen flows / interaction design / layout / information architecture) was that there was rapid feedback on implementation not only between the UX specialist and designer, but also for the end user, which promoted discussion and further refinement. The rework was a small price to pay for the overall quality achieved.

This reinforced a key ingredient for successful agile projects, to be responsible about UX design and tackle the areas of high cost of change. Put more effort around designing those, but keep in mind the benefit of rapid feedback and continual refinement and be prepared to revisit some areas of the site to apply a final "polish". Take time to understand the cost of change against the extra time spent in design by getting input from the rest of the team.

3.2.2 The importance of the product backlog

From the initial story-writing workshop the product backlog wasn't in a great shape. It had all of the scope in it, but there was little structure and the stories overlapped. One of the first activities was to get some structure on the backlog, by adding themes [G2] to group stories together. This helped to identifying overlapping stories within each theme. Another way to look at themes is to think about groupings of similar functionality, epics (large user stories) can also be used to express a broader amount of functionality.

One of the advantages of having the backlog structured is that it is suited to the generation of, and reference to, UX activities and artefacts. It also helps to support discussions with stakeholders concerning “richness” of functionality in each of the functionality areas, also known as a ”Minimum Marketable Feature” [G3]. I have found communicating with stakeholders at a theme or epic level, rather than a story level, is more effective.

The example below shows a simple theme called “create book record”, a simple UX sketch has been drawn to illustrate what create book record could look like, and thought has been put into the layout and flow of creating a book record. The diagram illustrates how each of the stories in the theme relate to the create book record sketch.

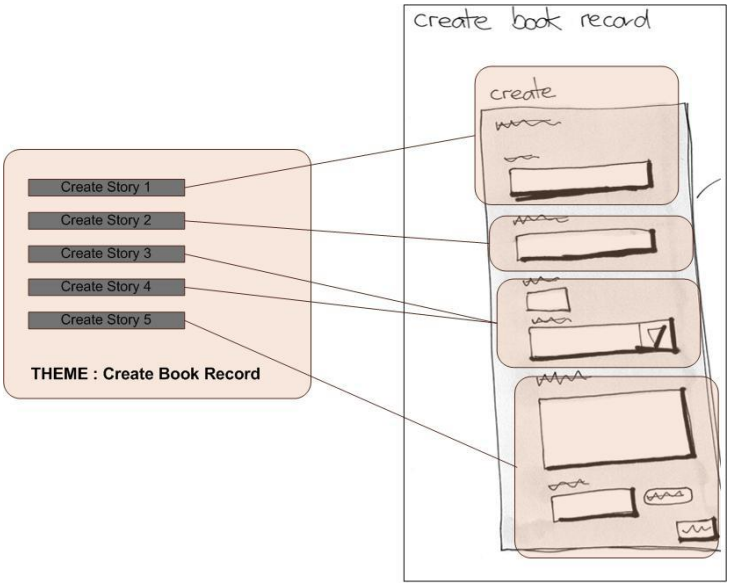

Figure 1

The team initially didn’t put much thought into how to prioritise the backlog to support the build order, and the generation of UX artefacts. Once there was some structure then we could communicate at a high level roughly what order the themes might be built in, to support generation of UX artefacts prior to development. This helped the stakeholders understand when certain functionality could be expected and allowed them to focus their discussions and ideas for interface design and functionality prior to the sprint planning meeting. This resulted in a more streamlined planning process that focussed on getting the required detail to support the development of the product, with associated sketches and concept design artefacts to “bring to life” the discussion in planning.

Agile projects benefit greatly from the use of mock webpage designs, however there should be consensus that they are there to support and help the discussion, rather than an accurate representation of what the developed page will look like. There was a great deal of benefit having these artefacts to reference in the sprint planning meetings. This is especially true in the early stages of product development or new feature development.

Once the product backlog was in good shape we could support feedback on the solution. Where any gaps or issues were identified we could either resolve them in the sprint if they were small, larger prices of work went on to the product backlog.

3.2.3 Agile mindset

From previous experiences I understood the impact of having a stakeholder not bought into the concept of using the product backlog. If the main budget holder does not buy into the fact that scope has to be flexed in order to meet deadlines and budget then agile projects get very difficult.

The product backlog then becomes a status report; outlining how much is left to develop, based on initial expectations. Instead the product backlog should be used as a tool to discuss and collaborate with stakeholders in order to make joint decisions concerning project direction.

I understood Adam hadn't bought into this concept by observing this behaviour in one of the very first meetings, when he refused to arrange backlog items into "must haves" or "could do without" for an initial release (everything was a must have).

Once the product backlog was in a better shape the team spent more time with the stakeholders to get a shared understanding of the purpose of the backlog using workshops to get their involvement. Initially this was still difficult, but once Adam and the other stakeholders were more involved, they took more ownership of the backlog and we began to use it to make prioritisation decisions based on budget and priority.

This doesn't directly relate to UX, however once the backlog is being used as it should be then different options for the implementation of usability features can be explored. By using story points [G1] to cost implementation options the stakeholders can make decisions on options with a broad understanding of their cost (time and money). This relies on using the backlog and previous sprint performance to make more informed decisions on implementation.

3.2.4 Frequent reviews and playback

Minimising the time between initial discussions of a feature, designing it, implementing it, reviewing and testing it was really important. This team achieved short feedback loops a few sprints in, the result being the stakeholders got to feedback more rapidly, the reviews were easier and the quality much higher – reflected by a happier team, stakeholder group and very few bugs.

This was achieved by ensuring that the whole team (customer, users, analysts, UX, designers, developers, testers) focus on smaller pieces of functionality and build them as quickly as possible.

The whole team was involved in the definition of a feature. Prior to a sprint starting solution options for product backlog items were explored, usually through discussions supported by sketches. Decisions on solution direction were then reflected in well defined product backlog items, supported by the sketches. Collaborative discussion tends to drive out solutions that are possible to develop and will allow you to explore options for more or less complex solutions.

Once a product backlog item is started within the sprint then it should be fed back to the users and UX as quickly as possible inside the sprint. Don't wait until the sprint review to gain customer feedback.

3.2.5 Stick to Scrum

It might be an obvious one, but sprint planning meetings aren't suited to drive UX workshop activities, they don't allow the team to get what they need to get out of the meetings, primarily planning backlog items by generating tasks and estimates.

UX specialists need to come to Scrum meetings with a view of adding a UX perspective to the product owner role; this requires effort prior to the sprint planning meeting. Use the Scrum meetings to add a UX aspect to the product owner roles to help the team understand how to implement features (in sprint planning) or feedback on features already built (in sprint review). Ensure that as a UX specialist you collaborate within the sprint for backlog items being developed.

3.3 What they would change next time

Having spoken to the team there were a number of areas that they would improve in the future. Some of these are outlined below.

3.3.1 Understand the flows through the system

If the user flows through the system are understood then these can be focussed on. Develop each flow, first building the path likely to be taken by most users then add on the exceptions.

This way the team could have driven out small usability inconsistencies that appeared as the site developed. Some of these areas were addressed in the initial release; others are still on the product backlog awaiting implementation.

There was a general feeling that the agile approach taken produced a better quality product without first having an initial UX detailed big picture, but focussing on the user flows would have generated an even better quality solution with less gaps.

3.3.2 Having a style guide in place as soon as possible

Having a style guide in place helped with the ensuring the overall consistency of the site. Whilst it may not be required immediately for initial sprints there may be some rework required as the team explores slightly different ways of approaching solutions, leading to inconsistency across the product. The style guide for this team appeared late in the project.

Once the style guide had been produced it was invaluable for reviewing areas of the site already built and ensured they were consistent. The additional work required was made visible by placing it on the product backlog and prioritising it with all of the other work. Once available the style guide was added to the team's definition of "DONE" i.e. what needed to be completed in the sprint for all backlog items developed.

3.3.3 Visual assets required in sprints

Having the visual assets (images for use when coding the page) available for use in development from the beginning of the sprint, or very early in the sprint will limit rework or unnecessary hold ups within the sprint.

4 Project Three

4.1 Overview

This team was split across two organisations; a large organisation "the customer" and a smaller development agency "the supplier". The customer had decided to use an agile approach to enhancing an existing website.

I was introduced to the project relatively early on in project life cycle; the initial requirements had been generated but no development work had yet started. This presented the opportunity to ensure that we set a clear vision for the project, and put in place a good structure on the product backlog using themes.

We also spent some time understanding how much design work was required for some of the new feature areas in order to prioritise the work. We took into account technical and design complexity in order to generate a prioritised product backlog that reflected a "build order". This allowed time for the more complex UX design areas to be explored prior to being built.

The customer had a brand sign-off process for public facing websites. They were extremely protective about their external facing brand and had put in place a Brand Review Board that met weekly and were used to dealing with high fidelity UX artefacts, so allowing time for developing UX artefacts in order to go through this process was key to ensure that the Brand Review Board had visibility and accepted any proposed designs.

Early in the project we put together some high level estimates (story points) [G1], for the work outlined in the product backlog. Whilst they didn't give exact estimates for when items could be delivered the estimates provided some guidelines as to when the designs would be needed in order to build the solution.

These estimates, along with an immediate start to developing some areas of the project, meant that the team had to look at ways of reducing the time required to specify and gain sign off for UX artefacts, in order to meet the more rapid software build times they were experiencing.

Jo was the business stakeholder and was responsible for gaining sign off from the Brand Review Board; she used the backlog to identify areas of high impact design change and those of little or no impact. The brand signoff process usually reviewed all design changes, but Jo looked across the backlog and discussed with the UX designer the impact of each of the areas of change and made a call on the severity of that change.

For high impact changes the team developed lower fidelity artefacts than were usually delivered. They were sufficient to share ideas, their acceptance by the board was supported by the fact that the final signoff was a demo of the developed site. This approach also allowed Jo to get "final signoff" on the lower impact changes that were only viewed by the board during the demo. This saved design time and effort for developing UX artefacts on the project.

The estimates on the product backlog also allowed Jo to look forward and make decisions on what features could make the final release. This forced the team to work collaboratively to come up with more cost effective solutions than had been proposed in some design artefacts. They found that forcing the UX designer, developer and Jo to work on possible solutions for upcoming backlog items they arrived at simpler functionality at a lower cost, without impacting the usability of the solution. Without a well kept backlog with story point estimates this approach to forward design may not have been attempted.

The project completed with a satisfied Brand Review Board, 50% quicker than similar projects, with a comparable level, if not better, design quality and didn't go over budget. This meant both Jo and consultancy provider were happy.

4.2 Observations

This project had been able to benefit from experiences in previous projects so they had solid agile project foundations in place; a good product backlog, well written product backlog items and an understanding by Jo that the product backlog was a tool to make scope decisions and not everything on it was achievable within the budget and timescales.

The team didn't have to spend time and energy focussing on getting the basic Scrum process working; they could instead focus on the value added areas of team work and collaboration across the entire project structure.

There were a number of observations that were identified in the initial sprints; some of these observations go towards answering common question on UX in agile projects, and steps taken to address those issues. These are outlined below.

4.2.1 Gap between UX and Development

The consultancy was not a large organisation and the developers and UX designers didn't sit far away from the developers. There was little bureaucracy, generally it was an open and inclusive environment to work in.

What was interesting is the behaviour that was observed in the first sprint review, within a team of four (three developers, one UX Designer). As the functionality was being demonstrated the UX Designer was feeding back on the areas missed by the developers, based on the original designs. It was obvious that there was no conversation between the UX designers and developers during the two week development, and all the feedback which was perfectly valid had to be captured and addressed in the next sprint.

This is typical of the behaviour that I experience on projects; many specialists are focussed on getting "their bit" right and not thinking how they can contribute to the whole.

Sticking to the Scrum process meant that we could identify this early and work on improving it, using the review and retrospective meetings. It was 3 or 4 sprints before we saw the collaboration we were hoping for, with designers and

developers reviewing software within the sprint. Forcing this collaboration also helped to support the upfront collaboration that followed for planning future backlog items to meet budgetary constraints.

4.2.2 Synchronise time spent on UX assets

Due to the involvement of the Brand Review Board there was a need for some upfront UX artefacts to be developed. The story point estimation that the team allocated to future stories on the product backlog helped Jo make decisions on scope implementation for each feature and delivery across features. Understanding how much time it would take to develop the rest of the product backlog in light of budget and time constraints, helped the team look for more cost effective solutions to implement functionality.

Nearing the end of the planned release some of the story point estimates hadn't been revisited and didn't reflect a common understanding of the backlog items. This meant that Jo expected to complete more work than was possible, so the UX artefacts for back log items not developed in this release had been created, resulting in waste.

Keeping all specialists on the team working together and focussed on a subset of the backlog is good practice. This allows them to benefit from refinement and adaption based on feedback. They should continually review future work against project resource constraints, this should minimise wasted effort on items that may not be developed.

4.2.3 Gaps in the "Product Owner Team"

Scrum advocates a single role for the product owner; given the breath of responsibilities for the role I rarely find all those attributes in one person (I'm being polite, I have never found them all in one person, and believe me I do try). For me UX is also part of the product owner role. Treating UX this way ensures that I don't over complicate Scrum and can still get across basic principles.

I often advocate a product owner team but with this implementation come challenges. This project had a three-person product owner team (a technical business analyst, Jo and a UX designer) with part of the product owner role also being carried out by the Brand Review Board. Jo still had overall responsibility for priority decisions but needed input from the others to get the right priority and gain sufficient definition of backlog items and accept the work done by the teams.

I have found on a number of occasions that having a product owner team that comprises technical, business and user experience skills sets is good enough to drive priority and get good collaboration and definition for product backlog items. It is important that this team is not set up as a committee, there should be a single person empowered to make decisions on functionality and prioritisation within the team. This person is usually from a business background.

Even with this small number of people there were a number of occasions where two of them would have a conversation, or one of them have a conversation with the development team without informing the others. On a number of occasions this caused confusion, because they were following Scrum they realised this either during the sprint review or sprint planning meetings. It was however an ongoing source of frustration, more for wasted effort on acceptance criteria and testing than anything else.

So just a warning, if you are going to have a product owner team make sure that you communicate frequently and synchronise at least weekly, I prefer to run a weekly product backlog meeting with all product owner team members present to ensure synchronisation. Also make sure that the product backlog is your one version of the truth; a product backlog management tool helps here.

4.2.4 Agile mindset

This keeps coming up; but unless stakeholders and teams are open to exploring different ways of working, you will make compromises with agile projects. This will impede your progress to minimising delivery times and maximising value through delivery of software in as short a time as possible. Anything that slows down that progress is creating waste in your organisation.

The interesting point with this project is that it was viewed as an innovation project, a project that within the organisation was subject to less control, although many other innovation projects did not take advantage of that fact. This meant that the suggestions made by Jo concerning changes to Brand Sign off Process met less resistance than other "main stream" projects may have met. This is the bigger challenge for organisations, in order to maximise the value chain for software development bigger changes outside setting up an "agile project team" need to be made.

This project made an initial step to ongoing change and was an ideal candidate for an initial agile project to highlight that a change in process may not necessarily reduce the quality of the output, and start to build some trust in the organisation to try more projects this way.

In Version One's 2009 annual agile survey [R4] one of the leading reasons for failed agile projects was "Company philosophy or culture at odds with core agile values".

4.3 What they would change next time

Having spoken to the team there were a number of areas that they would improve in the future, some of the higher priority areas were:

The estimates on the product backlog were a key focus for making decisions on how to approach the next stages of the development. The estimates were put together relatively early in the project, and not revisited often enough.

Next time the team would ensure that they continually reviewed those estimates and ensure that when they are generated they reflect collective thinking. Nearing the final sprints the stories were underestimated due to lack of information or changing requirements, which meant that prioritisation decisions would have been different.

5 Summary

I have provided an outline of experiences that have shaped the approach that I take on agile projects with a UX element. Similar approaches have also been used by others, as there are a number of published articles and papers citing similar approaches from a slightly different perspective.

So while these are not truly original thoughts here is some experience based advice outlining initial simple steps you can take on agile projects to help avoid a gap between UX, the customer and development teams.

My approach to coaching is initially to take the path of least resistance in order to gain some momentum. These steps reflect that approach as they refer to existing, widely used, UX and Scrum artefacts.

Some principles:

- Structure your product backlog
- Create a product owner team
- Prioritise the product backlog to support delivery as well as business value
- Make time to collaborate

There is benefit from using a structured backlog approach and use this to drive technical architecture, analysis, development and UX effort. Using UX artefacts to support the definition of epics and themes on a product backlog are an effective way to provide a UX perspective on a project.

There is a need to have a longer term view and spend more effort defining user experience for themes and epics where it is warranted e.g. for areas that have a high impact of change if they aren't well thought out. A well structured backlog supports this approach, with a product owner team working continually on the backlog to get backlog items "ready" [G4] for teams to be developed in sprints. This is turn will result in more predictable sprints, when using the

concept of ready teams typically have more accurate sprint planning estimates and complete whole backlog items in sprints.

When structuring the product backlog a use of clear release goals to identify a collection of MMFs helps to communicate overall project objectives to the team and stakeholders. Release goals can be used to define an overall roadmap, used in conjunction with story points and velocity you can estimate expected progress.

Use of a product owner team (comprising business, technical and UX skill sets) to structure the backlog, derive delivery order and reflect business value can be an effective approach. Business, technical and UX risks can be identified and reflected in product backlog priority. Options for themes, epics and stories can be worked through and efficient, usable, buildable solutions can be found.

If you do use a product owner team, ensure that they are part of the overall Scrum team and you don't get an "us and them" mentality, support collaboration and instil an agile mindset.

Whilst an agile approach may produce initial "imperfect" solutions each sprint, agile teams have experienced benefit from a more collaborative approach, not focussing on perfect design artefacts, instead sharing experiences and knowledge to derive solutions that fit broader project drivers. These resulting solutions may require iterating, further insight can be gained once enough of the product has been built to test and review user flows or journeys through the product.

There may be a need for UX specialists to create a "wow" factor for their work, either to build trust in ability or to win a pitch. If you do create polished artefacts initially for an agile project, ensure that stakeholders understand these are concept designs, as the project progresses these will be referenced and will provide creative direction for the product. They should not be used as an accurate representation of what the developed solution will finally look like.

Using Scrum to bring the stakeholders with you on the product development journey should mean that lower fidelity artefacts can be used to drill down into more detail, in specific areas. This approach worked well for Projects Two and Three and enabled them to avoid the possible pitfalls of Project One.

I will always push teams to minimise waste on projects, whether that is in time they spend on developing artefacts that specify functionality or time spent to agree on an overall solution before starting. Teams need to justify the time they are spending on getting backlog items "ready" by questioning the true cost of change of not getting the specification exact. Teams should also have a focus on continually improving, aiming to reduce preparation time and effort.

This is especially true of new projects; I rarely find a good reason why a product backlog cannot be prioritised to begin development in areas where it takes less time to get backlog items "ready". Other more complex areas are given more preparation time, by prioritising them lower on the product backlog.

The artefacts used in UX design aren't a bad fit for agile projects, and UX is an important part of product design. The main tensions I see for most teams are cooperation and understanding of what everyone can bring to the solution and an investment in time to ensure that they collaborate effectively to generate a workable, valuable, robust, efficient and usable solution as soon as possible, not just to "get their heads down and deliver".

Once teams begin to work this way, they find a natural level of documentation that supports their way of working. This would include the stakeholders. In my experience they, more than anyone, drive the need for a high level of fidelity in UX artefacts.

If you are on an agile project and stakeholders are requesting detailed, exact artefacts and not bought into using the product backlog, just be aware of the potential implications if you don't use a product backlog effectively.

5.1 A simple approach to get started

Lack of upfront planning is also one of the greatest concerns about adopting agile, and a barrier to further agile adoption [R4]. Outlined below is a simple approach that illustrates how widely used UX and Scrum artefacts can be used together. This will allow you to get some commonality between the UX and agile artefacts; its success relies on refinement and cross functional collaboration across the project.

This approach will give you an initial structure to your product backlog, it will help to identify areas of high impact or complexity from a UX, technical or business perspective. If you order the backlog based on these insights it will also provide you with some upfront planning ability for the areas that need it most.

5.1.1 Identify themes and group your Product Backlog Items within those

Use a story-writing workshop [R2] to get an initial product backlog, this approach will ensure that everyone has an understanding of the potential scope and a common understanding of any themes identified. A consistent, common understanding of theme definition is essential. A brainstorming approach and use of cards allows quick generation of stories, it enables everyone to assimilate and question the information and supports visible prioritisation.

You can either generate the stories then start to create themes to collect together related stories. I find that most projects have a good idea of what themes are involved in the project, so I sometimes start with themes and then generate stories for each theme. Once you have the stories arranged by theme, you can try and prioritise within the theme, arranging stories in priority order. An example output of a story-writing workshop is provided below in figure 2. For larger projects you may not get all these steps completed in a single workshop.

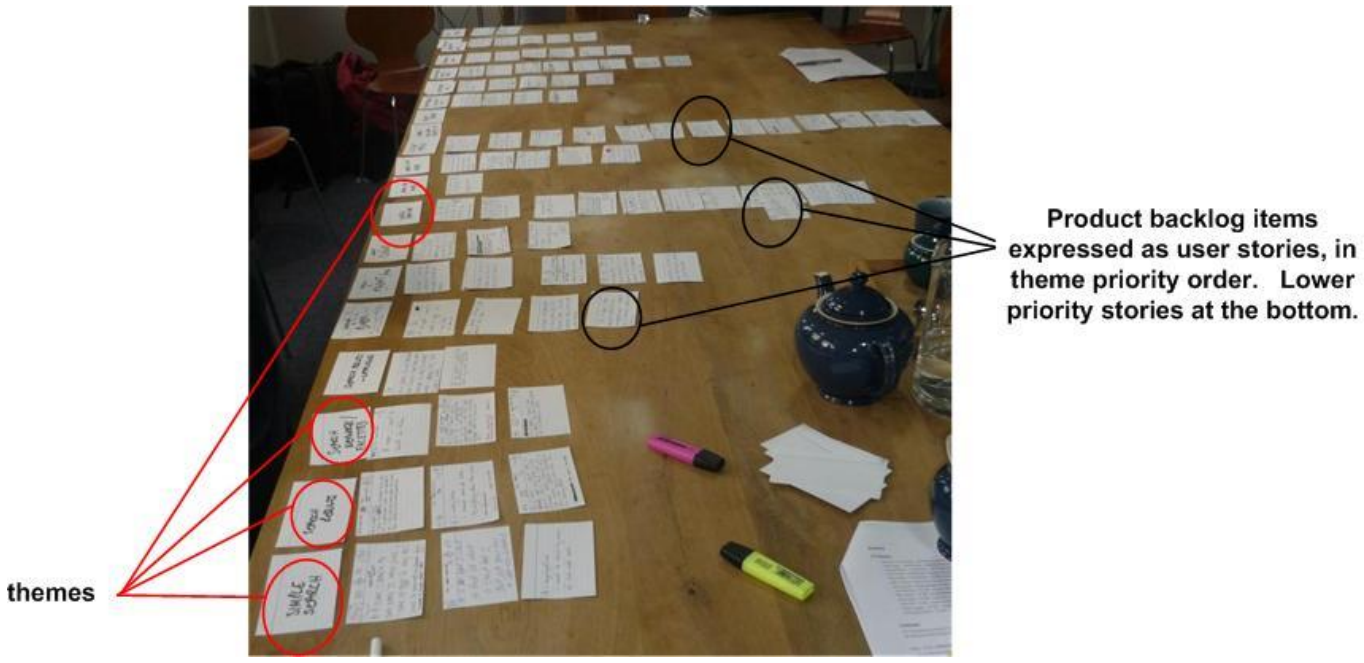

Figure 2

After that you may be able to create a release goal, draw a line below all the themes and put the stories above the line that would go towards achieving that goal and make a “minimum viable release” [G3], i.e. all the features that would be required to make a meaningful release to be given to the end users. Within each of the themes a minimum marketable feature will be identified. This is a very simplistic approach to identifying a minimum viable release and minimum marketable features, but it helps to focus on a release goal. If you have larger project divide it into a number of release goals, even if an initial release would not be given to end users.

Figure 3 illustrates a backlog that was generated with a focus on release goals. The product owner defined five release goals, each of these was written on different coloured cards. The product backlog items, expressed as user stories, that would make up each release were written on corresponding coloured cards, after the release goals were defined. This provided a really clear view of what functionality made up each release and which stories within each functional area (themes) would be developed.

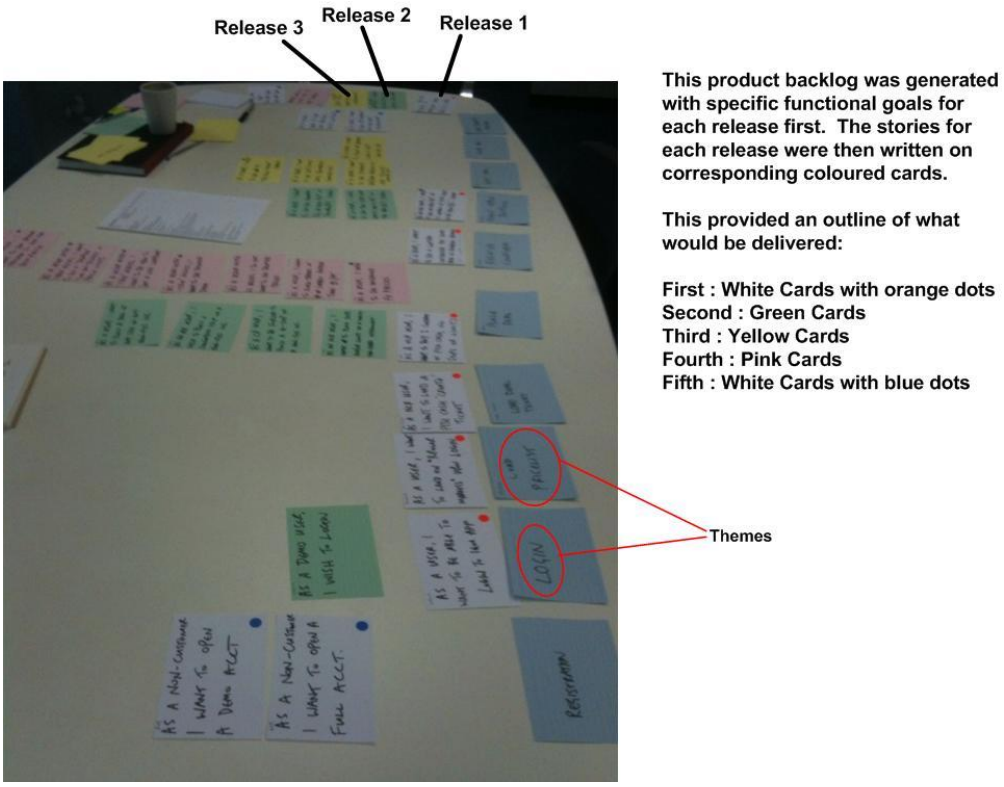

Figure 3

There are a number of different approaches that you can take to generate your structured product backlog, I would suggest that you use a workshop to brainstorm the user stories, attendees should include the product owner team and Scrum team. This will ensure that everyone has a clear view of what is being planned. From there you can use this to highlight areas of high cost of change (probably at a theme or epic level) and then generate a single prioritised list of all the product backlog items that reflects the delivery order.

This will then support the next step of creating UX artefacts to support those themes.

5.1.2 Create Some UX artefacts that support those themes

There may be one or more options for UX artefacts that support themes, find out the level of fidelity that fits your needs. If you have a wider group to gain agreement for a high impact design area you may want to be more polished, but with a preference for lower fidelity. The examples below aren't the only UX artefacts you may need, but are there to serve as an illustration. Figure 4 below shows different artefacts for a theme "managing a book record" and the "creating a book record" activity within that theme.

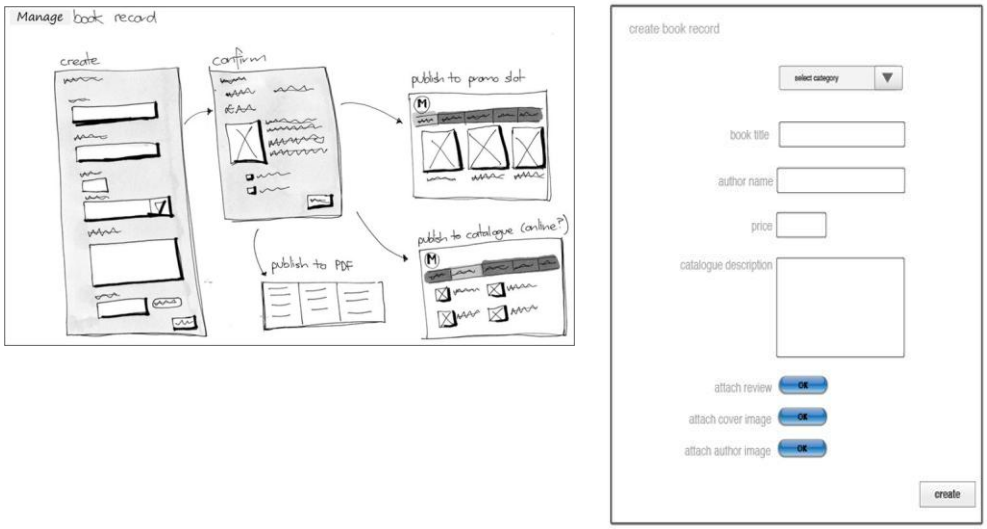

Figure 4

5.1.3 Have some mapping between the Product Backlog Items and the UX artefacts

Having this association and estimation on the product backlog items will help you make more informed product decisions. When exploring two or three options for implementation ensure that associated estimates for all specialists (technical, business and UX) are taken into consideration before deciding on the most appropriate approach. Story pointing each of the options will help you understand the impact of selecting a particular solution option.

Figure 5 below shows some simple mapping between the themes, sub themes, and user stories.

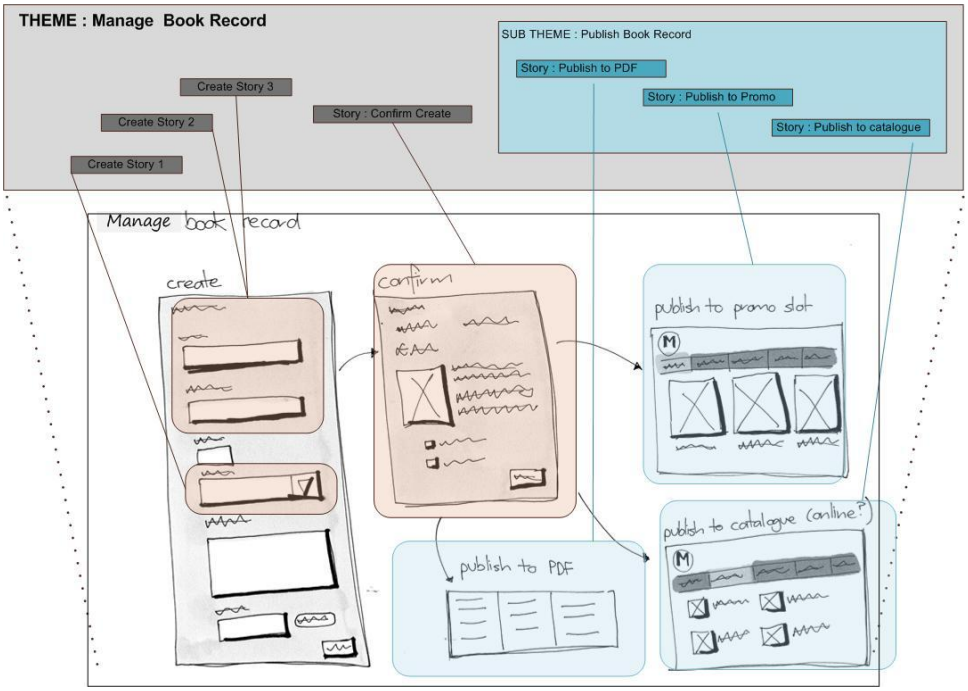

Figure 5

Using these simple steps should help you get started, there are a number of UX artefacts that I haven't covered here, but you may be able to support their use in this approach. Just don't forget to use the product backlog effectively, collaborate and take time to continually improve.

6 References

[R1] Ken Schwaber's Scrum Guide http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf

[R2] User Stories Applied by Mike Cohn http://www.mountaingoatsoftware.com/books/2-agile-software-development-book-user-stories-applied-for-agile-software-development

The User Story Workshop is also mention here: http://www.mountaingoatsoftware.com/topics/user-stories - When are they written?

[R3] User Story Map described by Jeff Patton http://www.agileproductdesign.com/blog/the_new_backlog.html

[R4] Version One 2009 State of Agile Development Survey http://pm.versionone.com/StateOfAgileSurvey.html. Barriers to Further Agile Adoption - Page 7; Greatest Concerns about adopting Agile – Page 7; Leading causes of failed Agile Projects – Page 7.

7 Glossary

[G1] **Story Points**

A technique to comparatively estimating product backlog items and associating a point value to each backlog item. We can use the point value assigned to product backlog items to gain an understanding of how many points we can take into each sprint, based on empirical data from previous sprints. This is known as calculating the team's velocity. Assigning story points across the whole backlog means that we can then use the velocity to understand how much work would could complete by a certain date or how long it could take to complete a planned amount of functionality.

Mike Cohn's Introduction to Agile Estimation and Planning

[G2] **Themes**

A term for collecting a group of product backlog items together, this is usually a functional grouping. This provides a useful way of referring to a group of user stories; it also helps in communicating to a wider audience.

The link below touches on themes, but also illustrates how they can be used to communicate with a broader audience.

Using themes to assist in visualising Product Backlogs

[G3] **Minimum Marketable Feature**

A term coined by Denne and Cleland-Huang in their book Software By Numbers (Pub. 2003)

"The smallest set of functionality that must be realized in order for the customer to perceive value. A "MMF" is characterized by the three attributes: minimum, marketable, and feature. A feature is something that is perceived, of itself, as value by the user. "Marketable" means that it provides significant value to the customer; value may include revenue generation, cost savings, competitive differentiation, brand-name projection, or enhanced customer loyalty. A release is a collection of MMFs that can be delivered together within the time frame."

Definition from Net Objectives http://www.netobjectives.com/glossary/

[G4] **Ready**

Time has to be invested in defining product backlog items in order to gain sufficient understanding and definition, especially in areas that have a high cost of change, if those areas need to change at a later date. Well defined backlog items also lead to much more predictable sprints, as they are easier to estimate and complete.

I have been using the concept of “well formed” backlog items since using it successfully on a number of agile projects, in order to add another dimension to product backlog prioritisation and allow appropriate time for UX and analysis activities. I presented these ideas at the Scrum Gathering in Spring 2008.

User Experience and Scrum- http://www.scrumalliance.org/resources/350 (slides 66 & 74)

Since then others have coined the phrase “Ready”, which to be honest is a lot catchier, so I’ve used it in this paper. I first heard it at Agile 2009 at a session presented by Jeff Sutherland and Carsten Ruseng Jakobsen. http://agile2009.org/files/session_pdfs/Going%20from%20Good%20to%20Great%20with%20Scrum%20Session.PDF

When a product backlog is first created having an initial browse across it will allow a product owner team to identify areas of high risk. This will allow the team to start to work on getting those areas ready, as they may take a significant investment in time. Used in conjunction with story points this can be a powerful way of understanding the impact of indecision or lengthy decision making on Scrum projects. Having this information can help the team identify and focus on minimising waste in this area.

[G5]Definition of Done

Done is an important concept in Scrum, one that is sometimes overlooked. The “definition of done” is a list of activities that should be carried out for each product backlog item that the team commit to each sprint. The Scrum team agree the definition of done with the product owner. This will ensure that everyone is clear on what is expected when the product is demonstrated in the sprint review. It also helps the team plan more accurately, the definition of done should always be displayed in the sprint planning meeting. Some examples of what a team may put on their definition of done:

- Group design to include product owner team and scrum team before starting a backlog item in a sprint
- Conforms to style guide
- Reviewed with UX specialist within the sprint
- Deployed and tested in the UAT environment
- Peer review