DSDM Agile Professional
Sample Synopsis for a DSDM AgilePF Project

DSDM Example Agile Professional Synopsis (for a project run under DSDM Agile Project Framework)

(NB.This is not intended to be a "perfect" synopsis or a "perfect" project! It is meant to demonstrate the level of information and topics that the examiners would hope to see covered by your synopsis. For example, the level of background information is minimal, since the synopsis focuses on the DSDM specific elements of the project. Where something does not comply with what DSDM recommends, it is helpful to know why you did it differently, and what impact this had, if any.)

Background

This project, for a government organisation, was to deliver a basic electronic Request and Authorisation solution in extremely short timescales and with a very limited budget. Since I had worked on a DSDM project previously as well as numerous rapid delivery projects, I was chosen as the PM.I also took on the Coach role – not ideal, but I was the only person with any previous DSDM experience.

The project approach

Given the very short timescales and the fixed delivery date (the project had to be live for the start of the new financial year – 11 weeks away!), I recommended we use DSDM, and since government organisations mandate the use of Prince, our project would need to be a combined DSDM/Prince project.

Pre Project stage

This DSDM phase is similar to a Prince Start-up. At this stage I was assigned as PM, and at a meeting with the Business Sponsor and the Visionary, the project governance, the project high level direction and the overall timescales for the project were agreed, the project was named and the initial budget was assigned for a short investigation and definition stage. Additional funding was reserved for the development stage, but was not yet committed. Since I had some experience of DSDM at a previous organisation and had previously attended DSDM Practitioner training, I suggested that the project should follow DSDM, and this was provisionally agreed. The decision to use DSDM was based on a review of DSDM's Factors Instrumental to Success

- *Embracing the DSDM Approach* - although DSDM had not been formally used in our organisation, a number of people had heard of it and in general supported a DSDM style approach. Ideally we should have sent team members on a DSDM training course, but the timescales made this impossible, and I felt that this could be addressed in the short term by an initial DSDM briefing session to explain the philosophy, the differences for this way of working, and the benefits for the business.

- *Effective Solution Development Team* – It was agreed that the team would be empowered to makes day-to-day decisions within the overall vision and strategy. Within our organisation, decision making and delegation is a fact of everyday life, so we agreed this was unlikely to present any problems, providing the boundaries of the team empowerment were made clear. We agreed to produce Terms of Reference for all team members, so that everyone was clear the extent of their empowerment. We all also agreed the escalation route for bigger project issues. (This worked extremely well during

the project, and ensured everyone felt comfortable with what they could decide and what needed to be escalated.

We also agreed the Solution Development Team size would be small (6 people) and that these would be ring-fenced for the life of this project (11 weeks). We also agreed that the team members chosen for this project already had the appropriate technical skills, but also demonstrated the right mindset and attitude for adopting Agile, as well as being good communicators.

We also recognised that some specialist developers would be developing modules against defined specifications, would not need to be closely integrated as team members and would therefore be handled as a separate group outside the DSDM model. We identified that the Architecture Integration team would have no involvement at a team level, but would be managed as an external resource via communication with the Technical Co-Ordinator. (This is not standard DSDM, but was made necessary by the formal approval structure and constraints of the organisation)

- *Business Engagement – Active and On-going* – Because of the high profile of this project and the short timeframe, the business agreed to commit significant time to the project for the 11 weeks, to ensure timely delivery of a working solution. We discussed the amount of time needed and balanced this against the fact it would be for a very limited timespan. We also agreed that meetings, workshops and user sessions would be put in people's diaries with as much notice as possible and these sessions would be sacrosanct.

- *Iterative Development, Integrated Testing, Incremental Delivery* – Given the short timeframe, there would only be one Increment delivered. However everyone agreed that using iterative development with testing integrated was the only way to meet the tight deadlines

- *Transparency* – the use of Team Boards, regular show and tells and retrospectives was explained, and the senior stakeholders were keen to see how this worked in practice.

Feasibility and Foundations (Combined)

Since the timespan for this project was extremely short (11 weeks start to finish), and since we had agreed to use industry standard technology building blocks, we decided to combine the Feasibility and Foundations. We kicked this initial phase off with a Facilitated workshop, with the aim of agreeing the objectives and outlining the technical approach and the timescales. We also discussed the life expectancy of the solution and agreed that we would need to maintain the solution from day 1. Out of this we produced a very short document which formalised the Feasibility of what we were doing. This would equate to a DSDM Feasibility Assessment. Once this workshop was complete, I then had enough information to produce an outline Delivery Plan. At this stage I also sat down with the Business Sponsor and the Visionary and we completed the Project Approach Questionnaire, and this helped populate an initial Risk Log.

At a meeting with the Visionary, we identified the Business Ambassador and a number of Business Advisors who had specific knowledge of how the process

worked out in the various regions of the organisation. Our Business Ambassador was a C Grade Civil Servant, responsible for the day to day Request process and the Head Office team. We then scheduled a requirements workshop to scope the requirements and agree our priorities. Since the organisation had no experience of workshops, I agreed to facilitate the workshop. Although this goes away from DSDM's recommendation to use an independent facilitator, I made it clear to all participants that I would be acting as the independent facilitator, and not as the PM. I found this quite difficult but it appeared to work reasonably well. As this was the first time the whole team were together, we also started the day off with a short session on DSDM and how it works. We focused on the DSDM Principles and how we would make them work, and I also explained the DSDM Practices of Workshops, Modelling MoSCoW prioritisation, Iterative Development and Timeboxes.

On the Principles, we came to the following agreement

Focus on the Business Need
We agreed that we would not necessarily deliver every requirement, but we would prioritise what needed to be done, and use these priorities to ensure the business would have a working solution meeting what they needed for day 1. We recognised that the basic business needs would be delivered by achieving our Must Have requirements, but we also felt that we would be able to deliver most or all of the Should Haves. We felt that we would be able to maintain focus on the business needs by having our Ambassador as part of our Solution Development Team and involved at least two or three times a week. He would also bring in the Business Advisors when they were needed. Our Visionary also agreed to attend key sessions (at the start and end of each timebox, as a minimum), and to respond to business questions within 1 working day, but quicker if possible. This gave the team a clear view of the up to date business drivers.

Deliver on Time
The whole team accepted that the end date was sacrosanct, and was not negotiable. Therefore if we hit problems, the business agreed that they would use the Prioritised Requirements List (PRL) to highlight requirements that could be dropped without compromising the end solution. Because we were working to such a short timescale, we all agreed to focus on a very basic and simple solution for this phase (Musts and Shoulds), and to allow workarounds as appropriate so we could meet the deadline. Some Could Haves, all Won't Haves, and any other comments and requests would be noted for a potential Phase 2.

Collaborate
Although we all felt the relationship between IT and the business was a collaborative one, we did discuss the sorts of areas where collaboration would need to be demonstrated, and we highlighted how this project was being treated as a joint venture, rather than a project requested by the Business and subsequently "owned" by I.T. This discussion helped set up an excellent working relationship from the start

Never Compromise Quality
Since this solution would need to be maintained from Day 1, we agreed the expectations for any features to be delivered into live, and we validated these expectations with our Support Team, to ensure they were also involved and aware of what was happening. This proved useful as they offered to provide some help with testing, and we all realised this would smooth the handover into Production at the

end of the project. We also agreed to use User Stories for our requirements, and use the pre-defined acceptance criteria, together with existing corporate technical standards, to agree whether a requirement was "done" or not.

Build Incrementally From Firm Foundations
We agreed that the requirements defined during the following workshop would be pitched at a high level, to allow us flexibility and to avoid getting pinned down in unnecessary detail in this early stage. All the users felt this was a sensible way to work. We also agreed that the PRL at the end of Foundations phase would be baselined, and that we would use subsequent Timeboxes to elaborate the detail and to build elements of the solution. I also highlighted to the whole project team that we would only be adding depth (through detailing the requirements) but would not be expanding the scope (the breadth).

Develop Iteratively
We agreed that early and on-going visibility of the evolving solution would be provided and that we would evolve the solution through a series of demonstration and feedback sessions.

Communicate continuously and clearly
We discussed how we would handle communication. Within the development team, we would aim for face-to-face communication wherever possible, with confirmations and decisions to be captured and circulated in simple e-mails. In terms of reporting outside the team, we agreed that progress reporting to the Steering Committee would be in terms of requirements "done", rather than percentage complete. We also agreed that we would use models and pictures, in preference to pure Word documents, wherever possible.

Demonstrate Control
We agreed that if the team were in control of their Development Timeboxes (and delivering at least their Musts, plus some Shoulds) then the overall Project would be in good shape. As PM, I agreed that I would use this information to demonstrate control of the Project to those outside our project. It was agreed that the Team would escalate immediately if they felt the Must Haves were at risk, so that we could make the best decisions as soon as possible.

The whole team agreed that discussing the detail of the principles was very helpful.

We then proceeded to have our Requirements Workshop, and following on from this we started updating the business elements of the Solution Architecture Definition, defining our Vision and Business Case for the project. We agreed our Prioritised Requirements List (using Must Have, Should Have, Could Have and Won't Have). We also flagged up the key non-functional requirements, prioritised them and added them to our list. These documents were circulated within two days for approval, and since everyone could see we had only written up what they had previously agreed, these were approved with no problems. As well as business people, the workshop was also attended by our Technical Co-ordinator, both our developers (as observers), our Solution Tester and our Business Analyst who acted as workshop scribe. Using the information from the workshop and in collaboration with the Technical Co-Ordinator, they updated the System Architecture Definition outlining the infrastructure and tools we would use on the project.

By this time we were a week into the project, so I organised a team session with the Technical Co-ordinator, the Developers, Business Analyst and Tester and the Business Ambassador, so we could agree a plan for how best to use the 9 weeks and confirm our estimates. This then formed the basis of our Delivery Plan. I also reassessed the project risks and reviewed the Project Approach Questionnaire with the group. At this point I was pleased to see that DSDM still remained the best approach. At the end of this phase, we had a Project Board Meeting where I was given approval to move into Evolutionary Development (in our organisation, this is a formal gateway review which must be documented before development can start) and our development funding was approved.

Evolutionary Development (working in Timeboxes)

At this stage, the team agreed to chunk development up 3 x 3 week timeboxes, allowing 1 week for Deployment.

Our initial timebox started with a short kick off and planning session for the solution development team – the Developers (one of whom took the Team Leader Role), Business Analyst, Tester and Business Ambassador, and an Investigation of the Functionality. I sat in at this session, but left it to the team to agree how they wanted to do things. From this, they produced their timebox plan. The 1st timebox would contain an initial combined perspective, focusing on Business and Usability, and would deliver several basic functional components. In our 2nd timebox, we agreed we would develop additional functionality, and our 3rd Timebox would develop additional functional components and also focus on the non-functional perspective. These were Timebox headlines, but we did not plan the Timebox details until the start of each timebox. At the planning session for each timebox, the team pencilled in dates for some on-going show and tell sessions, as well as a final Close Out session. At each of these review sessions, a Timebox Review Record was produced, in line with what we had agreed beforehand to be acceptable i.e. a simple e-mail, or Word bullet list, or a printed screen dump. Given the short timescales, focus had to remain on producing minimal documentation and using any techniques that would save time and allow the limited time to be spent on development. This was a very busy time for the team! In each timebox they evolved the Solution, together with appropriate test scripts, test data and test results encompassing both Technical and Business testing information.

As the solution evolved, further developer and business testing was done (which I insisted was documented. The developers were initially unhappy about this, as they had hoped that the short timescales would avoid the need to document their tests. But I explained the need for Repeatable Testing (a DSDM testing concept) and they agreed to produce Test Records.) This gave everyone growing confidence in the Evolving Solution which was baselined at the end of each timebox.

Because of the very short timescales, the complex internal testing procedures for production and the new technology, it was agreed that it would not be possible to do a stage release into a live environment, but that a training area would be made available for demonstration to the wider user community at the end of timebox 1 and each Timebox thereafter. Therefore the Delivery Plan focused on a single production release.

During Evolutionary Development I continued to assess the risks and update the Risk Log on a regular basis.

Deployment
Assemble and review - Pre-Deployment Testing

Although DSDM stresses the need for integrated testing, preferably testing the end-to-end experience as early as possible, this project had a specific issue in that the solution had to be formally tested and approved by an external Architecture Integration group before it could move into production. Assemble and Review activity within Deployment. So even though we had a Tester assigned from the Support Group, and although we had documented proof that the users and developers had tested all the way through development, we still had to hand the system over for a separate test phase before Deployment. This is not the route DSDM projects would normally follow. But the Architecture Integration Group commented on the quality of our testing and they were surprised this had been achieved given the short project timescale.

Deploy - Since this was a web solution with an expectation of minimal user training, the project focused on producing user documentation in the form of simple on-line help, which the Business Ambassador wrote. Once this was made available as a Production system, we had a review to assess what we had achieved, what had been left out and where we went from here. It was agreed to wait for a few months to allow the solution to bed down, before starting on Phase 2. In terms of DSDM products, we had a Deployed Solution and our review was documented in a Project Review Report.

Post Project

Once Deployment had been achieved, the project was handed over to the Support Team. At this point in time, we have not yet formally assessed what benefits have actually accrued from this project. This assessment has been scheduled for 6 months after our go-live date.

Roles, Responsibilities and Team structure

The Business Sponsor for this project was very senior within the organisation, the budget holder and although a busy person, she agreed to make time available outside the scheduled Project Board Meetings to deal with escalated issues.

The Visionary was also a senior person. He had an excellent view both of the way forward for the organisation as a whole in terms of the business, but also of the future aims for technology progress. He also agreed to give time to the project for regular review sessions and to deal with any escalated business issues.

The Technical Co-ordinator proved a key role on this project, since the architecture within this organisation is extremely complex and far-reaching, and he provided invaluable knowledge on what was possible/not possible and also for chasing for external resolutions for architecture problems and issues.

The diagram below shows the team structure we agreed upon. The relationship between the project and our Business Sponsor needed to be kept fairly formal, so all communications about the project were to be raised through me as PM. And

although the main channel of communication between the team and I was through the Team Leader (the direct line), I also wanted to keep informal communication with the team going, so I attended the daily stand-ups whenever possible, as an observer. Where necessary I would have a follow-up discussion after the stand-up, or would comment – but only if asked to do so by the team themselves. I was keen to ensure that the daily stand-up remained a team communication session, not a "report to the PM" session! The diagram omits the Project Steering committee. Communication to the Steering Committee was done formally, through standard Highlight Reports, Risk and Issues Logs. Progress was reported in terms of deliverables and requirements that had been accepted by the Business Ambassador.
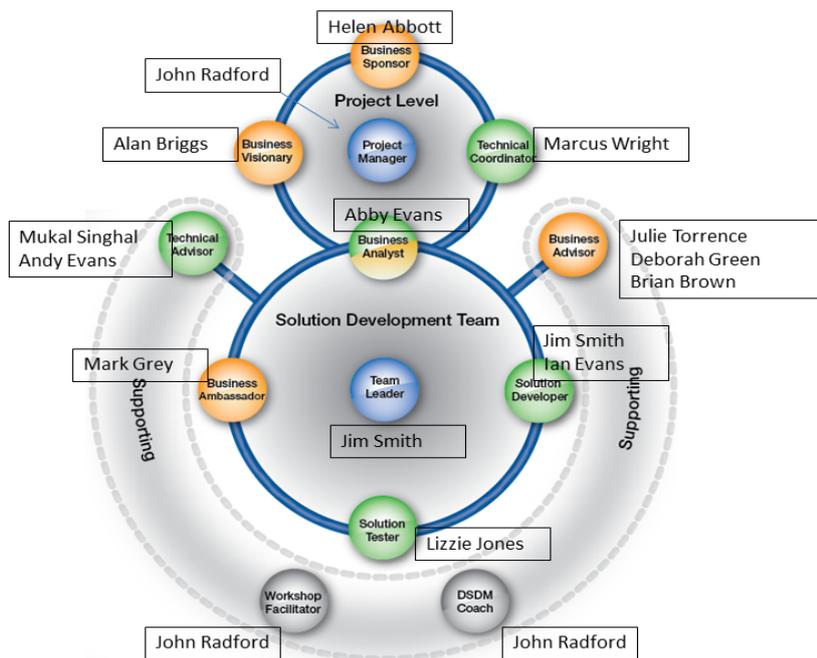
MoSCoW Prioritisation
For our business people, the idea of being expected to define the relative importance of their requirements was a new one, as they were used to just getting everything they asked for. We explained that they had a specific amount of time to get this solution working, so we needed to understand what was really important to them and what was negotiable, but initially they were very nervous, and tried to make almost everything a Must Have "just in case". To address this we kept reminding them of the definition of a Must Have "Without this one we will cancel the project and all go home". We also reassured them that we didn't intend to stop once the Must Haves had been achieved, and that if things went really well, then they might get all the Musts, Shoulds and Coulds, although this would be the best possible case. Over the 3 Timeboxes, the Business began to trust this way of working better.

Timeboxing
Working in short 3 week timeboxes, with the objective of banking working and tested functionality at the end of the Timebox was a new concept. But the whole team agreed to "give it a go", and everyone found it worked well. We agreed that with the amount of business time being committed to the project that we would use a Free

Format timebox, but would also agree days and times for discussions and show and tells with our Ambassador.

Iterative Development

For the developers, the process of Conversation followed by cycles of Thought, Action, Conversation was a very natural way of working. For the tester, this was a very different way to work, but after some initial hiccups, she found this really rewarding. The developers also realised that working closely with a tester is a real benefit, as between them they can have a very rapid test-fix-retest cycle. And now they wouldn't work any other way.

Workshops

Workshops proved a quick win on the project, as the ability to have an inform group agree the requirements and priorities for the project saved a lot of time, compared to the usual Interview, Document, Publish, Update, Republish cycle. And the retrospectives we held at the end of each Timebox helped us to constantly improve the way we worked.

Modelling

Our Analyst often used pictures (models) to demonstrate key concepts. What was interesting was when our developers and business people also started drawing pictures to explain what they meant – definitely an improvement on lots of large Words documents, and this avoids a lot of misunderstandings.

Lessons learnt / what I'd do differently next time

- We all felt we learnt a lot from having a Tester assigned to the project team. As well as helping the users, she also taught us a lot about what to test and why, as well as how to do the testing. We found the idea of benefit directed testing especially useful, given that we had so little time and yet still had to produce a production quality solution.

- All team members agreed that having clearly defined roles and responsibilities, together with a diagram of how the team was structured and how communication would be managed was very beneficial, and very different to other projects they had worked on. This is definitely something I will do in the future.

- Having a recognised (external) standard such as DSDM to follow helped the project a lot. If we were unsure, we could generally find guidance in the DSDM Handbook, and having this available electronically to everyone was very helpful. But for future projects I would definitely try to arrange some formal DSDM training at the start, for all those involved on a frequent basis (including the Business roles). Without this, on occasions we found we had different understandings of what we meant – especially around the area of what is a Must Have!

- Prince and DSDM – I had been told these would not work together, but this is untrue. I found that using Prince as the overall project management approach reassured the senior management and ensured an element of familiarity in the new DSDM environment, which helped build additional confidence. I also

found that it is very straightforward to produce "light" Prince products, which meet both the demands of Prince and the constraints of DSDM's rapid delivery.

Summary
Although this was a very hectic and stressful project, it was also very rewarding to work on something that delivered on time and met the business needs. The feedback we received from the users was very positive, and DSDM is definitely an approach we will be using in the future.