The DSDM Agile Project Framework for Scrum

www.nlightentraining.com
Andrew Craddock

- Partner at nlighten – www.nlightentraining.com
- Specialist in Agile Transformation and Training
- Technical Director of the DSDM Consortium
- 15+ years as an Agile Practitioner
  - DSDM Advanced Practitioner
  - One of the first Certified ScrumMasters in the UK
- 10+ years as an Agile Trainer, Consultant and Coach
  - Working with global brand leaders in many industry sectors:
    Airline, Life and Pensions, Investment and Investment Banking,
    Insurance, Engineering, Media, Mobile Telecoms, Pharmaceuticals
Popular Agile Approaches

- **Scrum**
  - Team focussed
  - Light, empirical Agile

- **eXtreme Programming (XP)**
  - Engineering focussed
  - Technical, disciplined Agile

- **DSDM**
  - Project focussed
  - Robust, scalable, governable Agile
DSDM Agile Project Framework

- An evolution of DSDM Atern
- Cast at a slightly higher level
- Designed to complement other Agile approaches
  - Lifecycle/Process more generic
  - Roles tweaked
  - Products completely reworked
  - Key Practices more obviously ‘recommended’ rather than implied as ‘mandatory’
DSDM Philosophy and Principles

**Philosophy**
- Projects aligned to clearly defined strategic goals
  - That everybody understands
- A focus on early delivery of real benefit to the business
  - A focus for or everybody

**Principles**
- Focus on the business need
- **Deliver on time**
- Collaborate
- Never Compromise Quality
- Build Incrementally from Firm Foundations
- Develop Iteratively
- Communicate continuously and clearly
- **Demonstrate Control**

DSDM key strengths
DSDM roles tweaked
- Allowing key Scrum roles to be substituted in
- Expanding on generic ‘Team’
  - Not as Agile but often a reality

Project level roles
- More senior, strategic view
- Direct and coordinate project
- Empower Development Team
- Keep out of the detail
- May act only via Product Owner if maintaining Scrum ‘purity’
What to fix?

Not everything!

The default for DSDM
Two distinct focuses
- Project Delivery lifecycle
- Evolutionary Development

Scrum Process slots right in
- Without change
- Delivering Potentially Shippable Product Increments
- Sprint by Sprint
Products Overview

Non-Scrum Products
- created where necessary
  - For governance reasons
  - Where they add value
  - To demonstrate control
Phase Objectives

To establish firm business foundations for the project:
- Business Case (Business Vision and Investment Appraisal)
- Product Backlog (Requirements ordered and prioritised)

To establish firm foundations for proposed solution
- Solution Architecture
  - Business Process / Business Organisation context
  - System Architecture framework
- Approach to development (incl. review & testing strategy)

To establish firm foundations for management
- Delivery Plan
  - Preliminary schedule of releases and ‘heartbeat’ Sprints
  - Will definitely change – probably after every Sprint
- Approach to Project Management and Reporting
**MoSCoW**

<table>
<thead>
<tr>
<th>Effort</th>
<th>Must</th>
<th>Should</th>
<th>Could</th>
<th>Won’t</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max 60%</td>
<td><strong>Must</strong></td>
<td>◆ Minimum Useable Subset</td>
<td>◆ Guaranteed</td>
<td></td>
</tr>
<tr>
<td>Min 20%</td>
<td><strong>Should</strong></td>
<td>◆ Work arounds difficult/costly</td>
<td>◆ Expected</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Could</strong></td>
<td>◆ Work arounds easy/cheap</td>
<td>◆ Bonus</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Won’t</strong></td>
<td>◆ Out of Scope this time around</td>
<td>◆ Maybe next time</td>
<td></td>
</tr>
</tbody>
</table>

- Cannot be de-scoped without causing the project to fail
- De-scoped as a last resort to keep the project on track
- Can be de-scoped without causing significant problems
MoSCoW & the Business Case

Cost based on 100% effort...

Must Have

Maximum 60% of total effort

Could Have

Should Have

Maximum 80% of total effort

Benefits based on Expected Outcome...

Worst Case Outcome

Expected Outcome

Best Case Outcome

Note: Business Case challenged by reduced benefit delivery rather than increased cost

© 2010-2012 nlighten ltd
Evolutionary Development using Scrum
Timeboxing and Compliance

2-4 (exceptionally 6) weeks

1. Sprint Planning & Investigation
   - 10% of effort
   - Agree Sprint Scope and priorities
   - Investigate detail of work to be done

2. Product Refinement
   - 70-80% of effort
   - Do as much of the work as you can in line with agreed priorities
   - Tie up loose ends, ensuring overall output of Sprint is fit for purpose

3. Sprint Consolidation
   - 10-20% of effort
   - Sign-off what has been delivered and assess impact of what has not

   1 hour Stakeholder Review checkpoints
   - Having thought about it in more detail... Are we doing the right thing in the right way?
   - Having done most of the work... Is it right yet? When not, what do we need to make it so?
   - Is it right? Yes or no...

Learn...
Inspect & Adapt ways of working
Sprint Backlog
- An agreed subset of the Product Backlog to be addressed in this Sprint
- An output of the kick-off meeting if following the DSDM Timeboxing Technique

Sprint Review
- A record of formal reviews of the evolving product (where valuable for governance/compliance traceability)
- An output of at least review point 3 if using the DSDM Timeboxing Technique

Technical Quality Control Records
- An output of all technical testing and review activity

Business Quality Control Records
- An output of all business testing and review activity
The purpose of the deployment phase is multi-faceted:

1. A final ‘assembly point’ for the solution – bringing together the business and technical aspects of the change and, where applicable, the potentially shippable product increment outputs of multiple teams.

2. A final check point for the integrity of the overall solution – which may include final testing in a controlled production-like environment (≠ traditional UAT).

3. The essential ‘go/no go’ governance decision point.

4. An opportunity for retrospection – similar to a Sprint retrospective – but for a Release.

5. As the phase in the project when the solution will actually be migrated to live use covering:
   1. Technical aspects of systems deployment.
   2. Training of end users and support staff.
   3. Required business organisation and process change.
   4. Formal handover of the solution to operational support staff.
Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **People and Interactions** over Processes and Tools
- **Working Software** over Comprehensive Documentation
- **Customer Collaboration** over Contract Negotiation
- **Responding to Change** over Following a Plan

That is; while there is value to the items on the right we value the items on the left more.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas
The DSDM Agile Project Framework for Scrum

andrew.craddock@nlightentraining.com