Dynamic Systems Development Method, DSDM

A whitepaper by John C Goodpasture, PMP

Managing Principal, Square Peg Consulting, LLC

www.sqpegconsulting.com

November, 2009

Dynamic Systems Development Method

DSDM is an agile methodology for small teams that is process centric, seeking to provide repeatable project results while respecting the need for customer value delivered incrementally.

The focal point for a DSDM project manager shifts from the traditional emphasis on tasks and schedules to sustaining progress, getting agreement on requirement priorities, managing customer relationships, and supporting the team culture and motivation.

Jim Highsmith

Dynamic Systems Development Method, DSDM, is an agile method that has its roots in the work of an early leader, in this case James Martin, who is generally credited with invention of Rapid Application Development, RAD.[1] But unlike other agile methods, DSDM itself was developed by a committee drawn from industry and academia who were seeking a standard framework and process for RAD to promote efficiencies. The initial framework was published in 1995 and has been subsequently maintained and revised by the DSDM Consortium.[2]

Somewhat like Feature Driven Development, FDD, and unlike the other agile methods that only loosely embrace process, DSDM is very process-centric, thereby giving guidance to project managers about how to get started and how to move deliberately to a successful finish. In fact, DSDM has a very specific multi-step process definition, with relatively fine grain steps.

For instance—and unlike all other agile methods—DSDM formally embraces business assessment of the project leading to a business case decision; then, like FDD, DSDM moves quickly to architecture and domain modeling. Process is supported by a robust set of practices, tools and artifacts, although managers are encouraged to shape the methodology to their particular circumstances.

Because of the robustness of the total framework, DSDM is probably the closest of the agile methods to the Rational Unified Process, RUP, supported by the Rational tool set and Rational's parent, IBM.[3] RUP is more practice-specific and tool-centric, giving more 'how-to' guidance than DSDM, but otherwise DSDM and RUP track well at the process level.

In addition to process, DSDM has a set of core principles, nine in all, that are the guidance for DSDM projects. In the same way that managers are encouraged to tailor processes to meet circumstances, so also they are encouraged to tailor principles.

There are a set of management practices, development practices, and process 'products' that are recommended as part of the framework. DSDM is not prescriptive about the exact form and content of the products, like the 'risk tool' or the 'time-box plans'; in other words, there are no templates provided by the DSDM Consortium. Rather, products are introduced and described, allocated to the process steps, but each project manager is encouraged to adapt the product specifically for their project circumstances. However, commercial tool vendors and professional

consultants do have templates to support each of the products.

And, of course, there is the team, the project manager, and the stakeholders. DSDM defines a role and responsibilities for each, in effect embracing the XP [Extreme Programming] idea of the 'whole team' as participants in the project community.

	Getting into DSDM
A Project Management Tip	• DSDM provides a checklist of prerequisites and initial conditions necessary for success • Because of its iterative process-centric format, flexibility afforded tool selection, and compatibility with traditional goals for repeatable project experiences, DSDM is a relatively easier methodology to adopt if coming from a traditional PDLC • There is much less emphasis than XP and SCRUM about letting developers interact directly with users to invent product 'on-the-fly'

The mind-set of DSDM is to deliver customer value on time

"Delivering on time" is a common refrain in the literature and practice of DSDM. To do so, DSDM embraces practices similar to other agile methods insofar as DSDM is incremental, not big-bang like a plan driven PDLC [project development lifecycle], and relies on time-boxes to regulate deliveries.

Like EVO , DSDM formalizes requirements prioritization. The requirements backlog is managed with a practice that has the curious acronym MoSCoW.[4]

Following the MoSCoW rules,[5] requirements are graded *'Must have'*, *'Should have'*, *'Could Have'*, and *'Want to have, but won't have this time'*. Each time-boxed iteration is planned to deliver some part of the requirements backlog, usually a mix of the 'must haves' and lesser priorities, but at least the 'must haves'; lesser priority requirements can be dropped during an iteration if necessary to meet a time-box constraint. The project as a whole is not successful unless all the 'must have' requirements are delivered; the system is not fully functional unless all the requirements, regardless of priority, are in production. Of course, the latter condition may not be economically possible, even though desired. DSDM embraces close-at-hand user interaction with the development team and feedback from operational evaluation of product increments. The up-shot is that the product backlog will change, as in all other agile methods. As the backlog changes, so will the priorities within the backlog. Thus, the must have count will be a moving target. However, there still must be respect for business milestones and investment limitations regardless of changing backlog count.

	On-time at milestones
A project management tip	• DSDM distinguishes itself with its overt acknowledgement that meeting a milestone is

	more valued than completing a requirements backlog. • Except for FDD, all agile methods embrace time-boxes for schedule control; all embrace backlog management to control scope, but DSDM, like EVO, 'overbooks' the iteration on the chance that more will be delivered if things go well

The process thinking in DSDM is iterative and incremental

DSDM borrows a page out of Winston Royce's iterative waterfall. There are five processes under the 'project' phase as given in Table WP2-1; the other two phases, neither of which has formally defined process steps, are *pre-project qualification* and *post-project close-out*.

In the 'project' phase, the first two processes, Feasibility Study and Business Study, are sequential, intended to be done only once. However, the subsequent three processes, Functional Model Iteration, Design and Build Iteration, and Implementation are intended to be iterative, to be re-visited more than once during a project as the user evaluates increments in production and offers feedback.

Table WP2-1 DSDM Processes

Process step	Commentary
Pre-Project Phase	No process step specified
Project Phase: Feasibility Study	• The opportunity is evaluated in the context of whether a project is the right solution and whether DSDM is the right methodology. • See Figure 2-2 for the top left side of the process flow 'V' model
Project Phase: Business Study	• Essentially build the business plan and the first iteration of the project balance sheet. • Determine resources, product vision, and general approach to implementation • Top-level requirements, technology needs, and other supporting data for a business decision are assembled
Project Phase: Functional Model Iteration	• Develop models, prototypes, and other artifacts to ascertain proper sequencing for required functionalities • Somewhat like a spiral, determine feasibility • Ultimately, form a consensus of a functional model of the product
Project Phase: Design and Build Iteration	• Design and build the final product • Agree on a release schedule to production
Project Phase: Implementation	• Train users • Deliver system • Assess satisfaction

Table WP2-1 DSDM Processes	
Process step	Commentary
Post Project Phase	• No specific practices • Develop a post-project plan for maintenance

Incremental is quite different from iterative. However, like its agile peers, DSDM is incremental as well as iterative. By incremental we mean that at certain scheduled release points, coordinated with the customer, product is put into production.

By incremental we also mean that at the end of every iteration, whether or not there is a go-live for the customer, product is finished and put into the product base. Of course, being incremental imposes a small tax on the project. Things stop and start, and at the stop all things need to be put away and made tidy. These are all efforts to be managed as part of the team's workload.

Every agile method faces these problems. As Jennifer Stapleton writes in her book, *"DSDM: Business Focused Development",* addressing incremental delivery is not uniquely a DSDM problem, incremental delivery having "…no distinct DSDM flavour."[6]

Principles guide day-to-day activity

The nine DSDM principles guide project managers and team members day-to-day. It is not doctrine that all nine principles must be present in every project, but certainly Principle 1 and 5 should be on everybody's list. In Table WP2-2 you will find the nine principles of DSDM.

Table WP2-2 DSDM Principles	
Principle	Commentary
1. There must be end-user involvement that is active and engaged	Like SCRUM, users are embedded in the team for best results
2. DSDM teams are empowered by decision policy to make their own decisions	Delegation to the lowest level is known as the principle of subsidiary function
3. Project management and development teams are committed to frequent delivery of product	Like all agile methodologies, the only measure of success that really counts is product delivery Planning for products is valued over planning activities
4. The primary quality measure is fitness for business use	Business use fitness captures the idea of usefulness, applicability, and effectiveness for the purpose intended
5. Development should be iterative and delivery should be incremental to ensure convergence to the best-value business solution	It is not possible to accurately, completely, and efficiently describe all the business requirements before some familiarity with the product is attained
6. All development changes to the product base are reversible	Configuration management systems and doctrine are applied so that changes to the product base can be backed out if necessary

Table WP2-2 DSDM Principles	
Principle	Commentary
7. Requirements developed during the business study are baselined at a high level in the business case	The high level baseline serves as the foundation of the domain model
8. Testing is built-in to the development methodology, not bolted on at the end	Testing is an everyday occurrence; testing occurs at many levels: unit, integration, system. Testing is primarily a verification activity to ensure compliance to design
9. Collaboration and co-operation among the project staff and with stakeholders is required for success	Collaboration for the greater good is the essence of teamwork

Management practices and process products empower the project manager

DSDM separates management and technical practices. The practices are supported by a number of process products.[8] Many are found within the other agile methods—time-boxing, for instance; others, like MoSCoW already discussed, are somewhat unique.

DSDM planning need not be too precise and prescriptive early on; planning should be 'just enough' just in time; and plans must be allowed to adapt and evolve. DSDM, suggests there be many plans: Feasibility and business studies, somewhat equivalent to a business plan; a time-box plan for each iteration, somewhat equivalent to the planning sessions described in the book; and various implementation plans to guide modeling, prototyping, building, and testing.

Project management is equivalent to managing the project balance sheet: communicating with executives, mentoring self-organizing teams, managing value earnings, and communicating with customers.

Risk management, as described by the DSDM Consortium is risk management 101, pretty much as described by PMI in their body of knowledge.

Measurements run the gamut from counting widgets, watching resource consumption, evaluating efficiencies, to the all important measurements for success.

For the latter, the DSDM method is value focused: is the customer satisfied; were requirements transformed into product; and were the must-have requirements delivered on time?

Quality is defined as 'fitness for business purpose' measured by customer satisfaction with performance, functionality, and features. Inspections and audits are allowed so long as they are not unduly intrusive and detrimental to efficiency and on-time performance.

Estimates are made for two purposes: to determine business feasibility of the project before it is approved; and to facilitate planning and 'control' during the 'project' phase. Most of the recommended estimating ideas are generic and found in any project management text.

Process products include models, scorecards, and plans of various types, nearly two dozen by count. As previously noted, templates and specifics are not offered since DSDM is considered an open

framework that can accommodate many tool sets and specific utilities.

Unlike other agile methods for the most part, there is a genuine commitment to documentation in the form of studies, models, and scorecards for management, and product documentation for the user.

	Project management choices
A project management tip	• DSDM offers project management a lot of choice so that the methodology can be shaped [note, methodology shaping is a Crystal concept] to circumstances • The framework is open; many products are suggested and described; principles can be flexibly applied and still be within the parameters for DSDM

Summary and take-away points

- Our theme for this discussion is that DSDM is an agile methodology for small teams that is process centric, seeking to provide repeatable project results while respecting the need for customer value delivered incrementally.

- DSDM is an open framework that supports a five step iterative process, with project managers and team members guided by a number of principles for project success. The methodology employs practices and products—tools, plans, and scorecards—to manage an iterative methodology that comports well with the Agile Manifesto.

- DSDM directs management attention to being on-time at customer milestones, delivering at least must-have functionality and lesser priority needs where possible.

- DSDM has a near-neighbor in RUP principles and practices; RUP is more tool specific and offers templates for plans and scorecards, but is otherwise largely compatible.

- DSDM is certainly a respectable member of the family of agile methods.

Endnotes

1 Martin, J. *"Rapid Application Development",* Prentice-Hall, Upper Saddle River, NJ, 1991. Prior to writing this book, Martin, from England, was a researcher at IBM.

RAD is a method, somewhat like the spiral, which relies heavily on prototyping as a means to begin development. It is 'light weight' in terms of documentation, and rather informal in terms of method structure. Hence, the motivation for DSDM as a more structured version of RAD and a more complete methodology.

2 The DSDM consortium is an industry-member organization, headquartered in England, that is responsible for the DSDM framework and supporting artifacts, www.dsdm.org

3 A comparison of DSDM to RUP is given in a whitepaper at dsdm.org:
Editor, *"White Paper: DSDM and Rational Unified Process (RUP*)", DSDM Consortium, 1999

4 The 'o's in MoSCoW are there just to make a 'fun' word out of the MSCW initials according to the DSDM consortium, http://www.dsdm.org/version4/2/public/moscow.asp

5 See MoSCoW Rules at http://www.dsdm.org/version4/2/public/moscow.asp, accessed August 2009

6 Stapleton, J. [Editor] *"DSDM: Business Focused Development – 2nd Edition",* Addison-Wesley, 2003, pg 22

7 See http://www.dsdm.org/services/faqs.asp, accessed August, 2009

8 For a complete description of principles, practices, and products, view the DSDM manual online at http://www.dsdm.org/products/dsdm_version_4_2.asp

About the author

John C. Goodpasture, PMP and managing principal at Square Peg Consulting, is a program manager, coach, author, and project consultant specializing in technology projects, strategic planning, project office operations. He is the author of books, articles, and web logs in the field of project management. He blogs at johngoodpasture.com, provides presentations at www.slideshare.com/jgoodpas, and his work products are found in the library at www.sqpegconsulting.com.