CODE PROJECT®
For those who code

home   **articles**   **quick answers**   **discussions**   **features**   **community**

help

Articles » Development Lifecycle » Design and Architecture » Methodologies

What Is DSDM?

Marc Clifton, J. Dunlap, 29 Sep 2003

★★★☆☆   3.09 (11 votes)

Rate this:

@   f   in   ✈   G+

A concise summary of the Dynamic Systems Development Method, one of the "Agile Methods"

About Article

Type	**Article**
Licence	
First Posted	**29 Sep 2003**
Views	**231,173**
Bookmarked	**48 times**

C++/CLI   Windows
.NET   Visual-Studio
Dev   Beginner

Contents
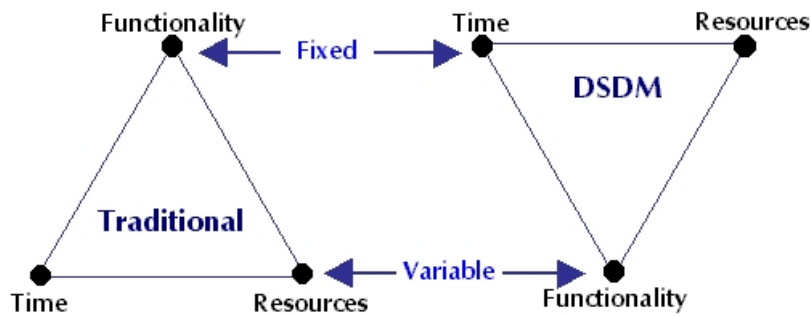
Introduction

Related Articles

Figure 1

Whereas in traditional development methodologies, functionality is fixed, and time and resources are variable, in DSDM, time is fixed, and functionality are variable.

Dynamic Systems Development Method (DSDM) is an organized, common-sense process focused on delivering business solutions quickly and efficiently. It is similar in many ways to SCRUM and XP, but it has its best uses where the time requirement is fixed.

DSDM focuses on delivery of the business solution, rather than just team activity. It makes steps to ensure the feasibility and business sense of a project before it is created. It stresses cooperation and collaboration between all interested parties. DSDM makes heavy use of prototyping to make sure interested parties have a clear picture of all aspects of the system.

Why DSDM?

- Results of development are directly and promptly visible
- Since the users are actively involved in the development of the system, they are more likely to embrace it and take it on.
- Basic functionality is delivered quickly, with more functionality being delivered at regular intervals.
- Eliminates bureaucracy and breaks down the communication barrier between interested parties.
- Because of constant feedback from the users, the system being developed is more likely to meet the need it was commissioned for.
- Early indicators of whether project will work or not, rather than a nasty surprise halfway through the development
- System is delivered on time and on budget.
- Ability of the users to affect the project's direction.

How Does DSDM Help to Overcome Development Pitfalls?

Many systems fall short of meeting the needs of the users and purpose they were designed for, causing the system to either be abandoned or overhauled. There are a number of ways this may happen:

- **Failure to meet the purpose / solve the problem it was designed for** - DSDM allows for user testing all through the development process, thus allowing developers to get prompt feedback on the usability and suitability of the product.
- **Cost outweighs benefits, or cost is too high altogether** - In DSDM, a Business Study is done at the beginning of the project, greatly decreasing the likelihood of late surprises in the financial realm.
- **Poor communication between involved parties** - DSDM stresses communication and collaboration between all interested parties - developers, users, etc.
- **The users finds the program either too hard to use that it does not work as expected** - DSDM allows for user testing all through the development process, thus allowing developers to get prompt feedback on the usability and suitability of the product.
- **Hidden flaws surface in the system due to poor design or implementation** - In DSDM, prototyping helps to ensure that the system is designed correctly and that everyone knows how it will work. Unit testing helps to uncover hidden bugs, and incremental development allows for user testing all through the development process.
- **Users resist the instantiation of the system, either for political reasons, or a lack of commitment to it** - In DSDM, since the users are actively involved in the development of the system, they are more likely to embrace it and take it on.
- **The system is in-flexible and/or un-maintainable, and unable to adapt to change** - Since DSDM emphasizes flexibility in design, this is not likely to happen with DSDM.

Related Research

Core Concepts in DSDM

The following bullet list describes the core concepts of DSDM.

- **Active User Involvement**

The people who will be using the product must be actively involved in its development. This important in order for the product to end up being useful to the people who will be using it.

- **The Team Must Be Empowered to Make Decisions**

The team should be able to make rapid and informed decisions, without having to cut through red tape to get those decisions approved.

- **Frequent Releases**

DSDM focuses on frequent releases. Frequent releases allow for user input at crucial stages in the product's development. They also ensure that the product is able to be released quickly at all times.

- **Iterative Development, Driven by User Feedback**

The development is the system is done in iterations, which allows for frequent user feedback, and a partial but prompt solution to immediate needs, with more functionality being added in later iterations.

- **Changes Must Be Reversible**

All products should be in a fully known state at all times. This allows for backtracking if a certain change does not work out well.

- **Requirements are Initially Defined at a High Level**

High-level requirements are worked out at the beginning of the project, before any coding, leaving the details to be worked out during the course of the development.

- **Fitness for Business Purpose is the Goal**

Meeting the business need is more important than technical perfection.

- **Integrated Testing**

Testing is done at every step of the way, to ensure that the product being developed is technically sound and does not develop any technical flaws, and that maximum use is made of user feedback.

- **Collaboration and Cooperation are Essential**

Collaboration and cooperation between all interested parties are essential for the success of the project. All involved parties (not just the core team) must strive together to meet the business objective.

- **20% / 80% Rule**

DSDM assumes that 80% of the solution can be developed in 20% of the time that it would take to produce the total solution. DSDM focuses on this 80%, leaving another 20% for later revisions. DSDM assumes that not all of the requirements for the final solution are known to begin with, so it is likely that the final 20% of non-essential features are likely to be flawed anyway. (See MoSCoW Prioritization.)

The DSDM Development Process
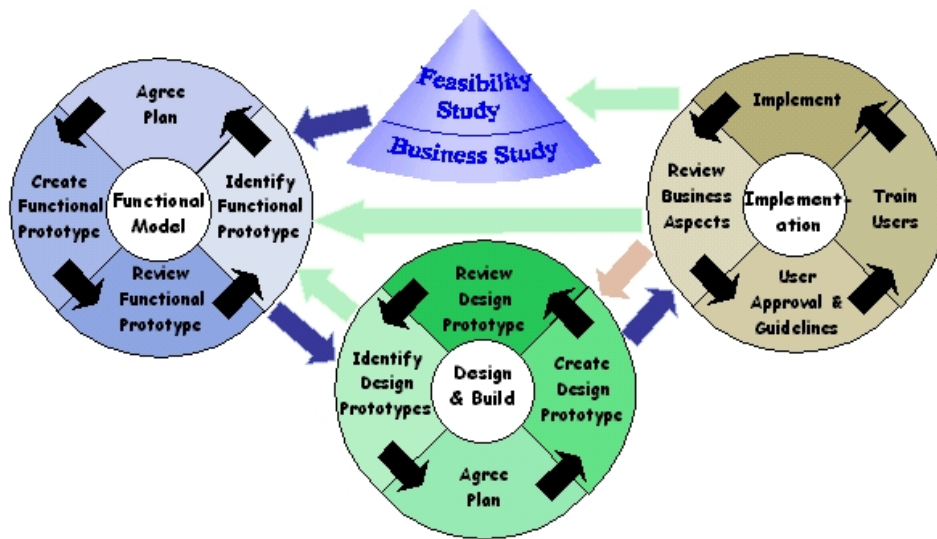
The DSDM development process consists of 7 phases. The first one is before the project has officially started. Then there are the project studies, which in this document are considered to be one phase. Then there are three more phases that consist of iterative cycles, which are repeated as necessary to complete the project. Then there is the post-project phase, where the project is maintained. The project flow may move between the different phases in the directions indicated by the arrows above.

The dark blue arrows in the diagram indicate the normal forward direction of project flow.

The green arrows indicate directions that may be taken as necessary under normal circumstances. For example, if the team has finished a "Design and Build" iteration, but the system cannot be released until another area's functionality has been defined and it has been built, the project flow may go back to the "Functional Model" iteration to complete that area.

The light red arrow represents a direction that is only taken if the project has been found to not meet the required functionality.

Pre-Project

The pre-project phase is not strictly defined. It occurs before the project officially begins. In this stage, the project is conceptualized, and the decision is made to start the project.

Feasibility Study - Finding out if and how the project will work out

In this phase, the team researches the question: Can it be done within the constraints of time and resources? This phase is done as quickly as possible, because DSDM is best used where time is short, and therefore the product needs to be delivered quickly.

Business Study - Studying the business aspects of the project

In this phase, the team researches the business aspects of the project.

- Does it make good business sense?
- Who are the participants and interested parties?
- What is the best work plan? What is needed to:

    1. Build it
    2. Test it
    3. Deploy it
    4. Support it?

- What technologies will we be using to build and deploy it?

The model is also worked out in this phase.

Functional Model
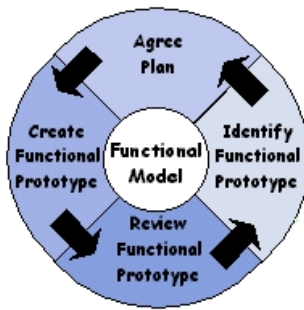
In this stage, functional prototypes of the system are made and reviewed. A functional prototype is a prototype of the functions the system should perform and how it should perform them.
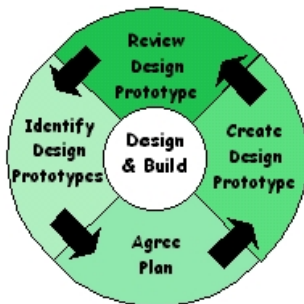
Prototyping follows these steps:

- Investigate
- Refine
- Consolidate

Prototypes cover many different aspects of the system:

- Business -
- Usability - how easy is it to use?
- Performance and Capacity - can it handle the volume and frequency of use that it will receive?
- Technique - What's the best way of going about solving the problem?

Design And Build

In this stage, the product is designed and developed in iterations. In each iteration a design model is made of the area being developed, and then that area is coded and reviewed.

Implement/Deploy/Maintain
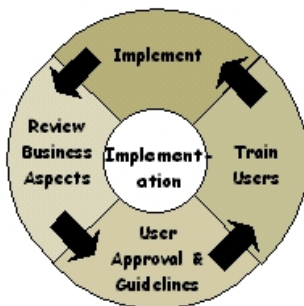
In the last phase, the product is wrapped up, documentation is written, and a review document is drawn up, comparing the requirements with their fulfillments in the product. The users are trained in how to use the system, and the users give approval to the system.

Post-Project - Maintenance

After the product is created, maintenance will inevitably need to be performed. This maintenance is generally done in a cycle similar to the one used to develop the product.

DSDM FAQ

Is DSDM suitable for all projects?

No. Even diehard proponents of DSDM agree that DSDM will not work for all projects. (But of course, nor will other methodologies.) More often than not, if full-on DSDM doesn't work, you'll end up using some DSDM concepts and principles, but not others. It is important to analyze the nature of the project at hand, and decide which concepts and principles will help you out, and which will not.

What are the basic criteria for the success of DSDM?

To use DSDM, there must be full management commitment, the team must be able to meet together easily, and the team must be able to work together easily.

How widely is DSDM used?

1 out of 5 developers in the UK use DSDM, and more than 500 major companies have adopted it.

DSDM Roles

DSDM defines several key roles that should be filled by members of the team:

- Ambassador - The person who acts as a go-between between the customer(s)/users and the development team. He coordinates the development team, and should have a good overall understanding of how the system will work.
- Visionary - The driving force behind the project - keeps the project steered on course towards the business goals. Often is the person who started/thought of the project.
- Advisers - People who have practical knowledge in area(s) of the business that need to be automated, and/or in the technologies needed to automate these areas.

There are also other roles that are defined by DSDM:

- Technical Coordinator - The technical coordinator coordinates the various technical aspects of the system, and ensures that they interact smoothly and correctly.
- Executive Sponsor - A strong proponent of the system, who has the ability to contribute funds and resources to the project.
- Project Manager - oversees the development and prototyping.
- Team Manager - Manages the people aspects of the team, and ensures that the team functions smoothly as a whole.
- Senior Developer - a senior developer who has the knowledge and expertise to turn the plans and requirements into deliverable code.
- Facilitator - Facilitates in the practical areas of the team's work.
- Scribe - The person in charge of documenting decisions, discussions, and plans made by the team.

Tools and Techniques

MoSCoW Prioritization

Functionality is categorized according to its importance:

- Must Have - the things that are absolutely essential, and are fundamental to the system.
- Should Have - things that are important for the business solution.
- Could Have - things that are useful, but can be done without for a short while.
- Won't Have This Time Around - things that can easily wait until later.

Prioritization is important because there is not enough time to do everything, and those things that are essential must be put before things that are not.

Prototyping

Prototypes are used heavily in DSDM. They ensure that all interested parties have a clear picture of the various aspects of the system.

Facilitated Workshops

Facilitated workshops allow for these benefits:

- Ideal environment for the formation of ideas and quick and balanced growth of those ideas.
- All interested parties are aware of decisions made by other interested parties
- Decisions can be made by a wider range of stakeholders.
- Decisions are made quickly and accurately.

TimeBoxing

Fixed time periods ensure that:

- Product delivery stays being the focus
- Concentration is on the top priorities
- All members know where each other is at and how long to expect each other to take

Links

DSDM Consortium - The official consortium for DSDM. Has a large number of articles and white papers on DSDM, plus DSDM certification, discussion groups, etc.

ProcessWave DSDM Articles

SurgeWorks DSDM Resources

Agile Alliance DSDM Articles - Articles about DSDM on Agile Alliance's website. Especially focuses on intermixing DSDM and other Agile methodologies.

DSDM Manual - A DSDM manual, with poor layout, but some good insights.

Articles

Introducing DSDM into an Organization

Top Ten DSDM Problems

To DSDM Or Not To DSDM?

DSDM Overview - A PowerPoint presentation outlining DSDM.

Books

DSDM: The Method in Practice by Jennifer Stapleton

DSDM: Business-Focused Development by Jennifer Stapleton

License

This article has no explicit license attached to it but may contain usage terms in the article text or the download files themselves. If in doubt please contact the author via the discussion board below.

A list of licenses authors might use can be found here

About the Authors

Marc Clifton

United States 🇺🇸

Marc is the creator of two open source projets, MyXaml, a declarative (XML) instantiation engine and the Advanced Unit Testing framework, and Interacx, a commercial n-tier RAD application suite. Visit his website, www.marcclifton.com, where you will find many of his articles and his blog.

Marc lives in Philmont, NY.

J. Dunlap
Web Developer
United States 🇺🇸

My main goal as a developer is to improve the way software is designed, and how it interacts with the user. I like designing software best, but I also like coding and documentation. I especially like to work with user interfaces and graphics.

I have extensive knowledge of the .NET Framework, and like to delve into its internals. I specialize in working with VG.net and MyXaml. I also like to work with ASP.NET, AJAX, and DHTML.

Article Top

Comments and Discussions

You must Sign In to use this message board.

Search this forum [_____] Go

☑ Profile popups   Spacing [Relaxed ▼] Noise [Medium ▼] Layout [Normal ▼] Per page

[25 ▼] Update

First Prev Next

📄 Thanks 📌	👤 BanksySan	14-Jan-06 8:00
❓ what is BDSM? 📌	👤 Amanjit Gill	2-Nov-03 3:59
📑 Re: what is BDSM? 📌	📧 Marc Clifton	2-Nov-03 4:24
Pareto Principle 📌		28-Oct-03 23:40

	ColinDavies	
Re: Pareto Principle	Marc Clifton	29-Oct-03 1:33
Isn't this getting repetitive?	**Stephan Meyn**	**28-Oct-03 17:51**
Re: Isn't this getting repetitive?	Marc Clifton	29-Oct-03 1:18
Re: Isn't this getting repetitive?	Stephan Meyn	29-Oct-03 10:27
Now I know ...	**Anonymously**	**6-Oct-03 2:44**
Do you sleep?	**Matt Gullett**	**30-Sep-03 14:28**
Re: Do you sleep?	jdunlap	30-Sep-03 14:34
Re: Do you sleep?	Marc Clifton	30-Sep-03 15:11
Re: Do you sleep?	Nishant S	30-Sep-03 16:07
Re: Do you sleep?	Anonymous	1-Oct-03 21:12
Re: Do you sleep?	peterchen	1-Oct-03 21:13
Re: Do you sleep?	Anders Molin	2-Oct-03 4:15
Re: Do you sleep?	Nishant S	2-Oct-03 4:37
Re: Do you sleep?	Anonymous	4-Oct-03 17:37
Re: Do you sleep?	jdunlap	4-Oct-03 17:41
Re: Do you sleep?	Anders Molin	4-Oct-03 23:39
Re: Do you sleep?	Jason Henderson	7-Oct-03 6:18

Last Visit: 31-Dec-99 18:00 Last Update: 14-Jul-14 1:43	Refresh 1

General   News   Suggestion   Question   Bug   Answer   Joke   Rant   Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.