DSDM®
CONSORTiUM

The DSDM Agile Project Framework

Handbook

www.dsdm.org

The DSDM Agile Project Framework

Disclaimer

This handbook, the DSDM Agile Project Framework, is intended for use as an aid to those undertaking DSDM training or revising for examinations in DSDM. Readers wishing to know more about or use the framework are invited to visit the website at www.dsdm.org

Note: all organisations offering accredited DSDM training must be accredited by the DSDM Consortium.

DSDM Consortium, Henwood House, Henwood, Ashford, Kent TN24 8DH, UK

Published by the DSDM Consortium

First printed June 2014

ISBN 978-0-9544832-9-6

Printed in the United Kingdom by:
Buckland Media Group Limited, www.buckland.co.uk

Acknowledgements

Editorial Team: Andrew Craddock, Barbara Roberts, Dan Martland, Dorothy Tudor, Jennifer Stapleton, Julia Godwin, Keith Richards, Mark Buchan and Steve Messenger.

Production: Emily Ruffle, Kim Whitmore, Mary Henson with artwork by Debbie Cole Creative Marketing

Contents

Foreword

Welcome to the DSDM Agile Project Framework Handbook produced by the not-for-profit DSDM Consortium. The DSDM Agile Project Framework is the leading, proven, Agile approach providing the governance and rigour along with the agility and flexibility organisations demand today.

The DSDM Agile Project Framework can be used either stand-alone or combined with other recognised methods such as PRINCE2™, MSP and PMI. It is also ideal as a wrapper for more limited Agile approaches to ensure that the whole project lifecycle is addressed.

The DSDM Agile Project Framework is a proven 'battle hardened approach' and has been responsible for the successful delivery of innumerable projects around the world. Its provenance across both IT and non-IT contexts goes back to 1994 with substantial productivity gains independently verified by the UK Software Metrics Association.

The DSDM Consortium operates on a collegiate model and this handbook represents the culmination of contributions from many experts over many years. Indeed, so many it is impossible to name them all individually here. However, in the spirit of Agile projects – you know who you are and on behalf of the Consortium, I'd like to send my heartfelt thanks to all of you.

This handbook is intended to support study and continued professional development in the field of Agile project management. It shows how the lifecycle can be applied to address the business need, mitigate risk and maximise the return on investment. As such, it represents essential reading for anyone wishing to become a certified professional as demanded by so many employers in today's market.

The content is based on what we describe as the DSDM Agile Project Framework core. That is – it dispenses with generic practice that can be readily found elsewhere and is not side-tracked by niche specialist usages. In this way, the handbook remains cogent and concise throughout.

However, information on specialist usages – along with contextual illustrations and a wide range of support material – is provided on the www.dsdm.org website. The majority of the DSDM website, including the DSDM Agile Project Framework, is free to view and free to use.

You may also wish to become a DSDM Consortium member. Quite apart from full access to all materials (including access to e-books) there are significant discounts on products, training and events. Furthermore, membership provides you with an opportunity to engage with the community and contribute to the framework – ensuring the quality and vibrancy of these past 20 years are carried into the future.

The DSDM Agile Project Framework Handbook represents the very latest in knowledge and learning around delivering project success and I am as proud of this product as I have been of the many others we have created.

I hope you find it both a helpful and rewarding contribution to your personal success.

Steve Messenger

Chairman

1. Introduction

1.1 DSDM and the DSDM Consortium

DSDM is a proven framework for Agile project management and delivery, helping to deliver results quickly and effectively and, over the years, has been applied to a wide range of projects - from small software developments all the way up to full-scale business process change. Although DSDM works easily and effectively on small, simple projects, it has always maintained a strong focus on the corporate project-based environment to provide a "grown-up" approach to Agile in the complex corporate world.

DSDM was initially created in 1994 through collaboration of a large number of project practitioners across many companies who were seeking to build quality into Rapid Application Development (RAD) processes as they developed, primarily, business-focussed computer solutions.

The contributing companies initially formed the DSDM Consortium as a not-for-profit organisation to manage the sharing, exploitation and evolution of the intellectual property of DSDM. Initially this was on behalf of, and exclusively for, the Consortium members. However in 2007 with the full support of its member organisations, the DSDM Consortium made DSDM universally available on a free to view and free to use basis.

1.2 The DSDM Agile Project Framework

The DSDM Agile Project Framework is an evolution of DSDM Atern®, the previous version of DSDM. It provides the information that is essential to enable any role on a DSDM project to use DSDM effectively and to understand how it is applied in practice.

In addition to the DSDM Agile Project Framework Handbook, the DSDM Consortium provides additional, more specialised information based on specific areas of interest via pocketbooks and handbooks, and white papers e.g. The Agile PMO. Please refer to www.dsdm.org for further information.

The framework retains DSDM's project-focused principles, together with its rich set of roles and responsibilities that are ideally suited to a corporate project environment. It also continues to embrace the same robust and fully Agile DSDM practices for establishing and demonstrating control in a project. At the delivery level, the DSDM process has been simplified to better reflect current trends in evolutionary solution development. The most significant change is in the area of the products, or deliverables, associated with the process. Although the level of formality and indeed inclusion or exclusion of individual products has always been discretionary, the latest product set has been streamlined to align more obviously with the Agile philosophy to keep essential documentation lean, timely and valuable.

From the project perspective, DSDM advocates that projects should do just 'Enough Design Up Front' (EDUF) within a Foundations phase in order to understand and clarify the structure of the overall solution and to create an Agile plan for delivery of the project. This puts in place the foundations for successful development and delivery, and is seen as a key differentiator for DSDM. It is important to understand that the Foundations phase of a DSDM project is very different from the analysis and design steps used in a traditional 'Waterfall' approach. In a DSDM project, analysis and design activity undertaken in Foundations covers the full breadth of the project but deliberately avoids going into detail (the depth). Substituting traditional 'Big Design Up Front' (BDUF) with DSDM's EDUF promotes agility in developing the required solution whilst avoiding the risk of 'No Design Up Front' (NDUF) that makes many larger and more strongly governed organisations nervous.

Where required, DSDM can be tailored, for example to complement other project management disciplines such as PRINCE2 or PMI, to avoid conflict or duplication of effort. It is also designed to complement Agile product delivery approaches such as Scrum that do not already have a full project focus.

The DSDM Agile Project Framework pocketbook provides an overview of the framework. The DSDM Agile Project Framework for Scrum white paper and the Agile Project Management and Scrum pocketbook offer a practical way of making the use of Scrum easier, more effective, or simply more accessible to project and organisational stakeholders in larger or more complex corporate environments who are already invested in the Scrum approach to delivery.

1.3 This Handbook

This DSDM Agile Project Framework Handbook represents the current guidance for DSDM applicable to all roles on a project, whether their interests are focussed on business, solution, management or process. It is intended to be used as a general reference for DSDM, as collateral for accredited training courses and for revision purposes when taking DSDM examinations. There is an on-line version of the same core material available at www.dsdm.org that is **free to view and free to use** by individuals within their own organisations, as well as a growing catalogue of extended guidance in pocketbooks, case studies and white papers dealing with optional depth and detail on how DSDM and/or specific aspects of it have been applied to projects and combined with other methods, or from a specific role perspective.

The chapters in **the first part of this handbook** provide a detailed explanation of the basic elements of the DSDM Agile Project Framework approach

- Choosing DSDM as your Agile Approach
- Philosophy and Fundamentals
- Principles
- Preparing for Success
- Process
- Roles and Responsibilities
- Products
- Key Practices
  - Workshops
  - MoSCoW Prioritisation
  - Iterative Development
  - Modelling
  - Timeboxing

The chapters in **the second part of this handbook** provide additional detail for the DSDM Agile Project Framework in practice

- People, Teams and Interactions – the key factor for successful DSDM projects
- Requirements and User Stories – best practice guidance for Agile requirements
- Planning and Control – how to control a DSDM project in a style appropriate for Agile
- Tailoring DSDM – some examples showing how the DSDM framework has been tailored for different circumstances

The Appendices provide supplementary information to support DSDM

- Appendix A – Glossary of Terms
- Appendix B – Project Approach Questionnaire

For more information, contact the DSDM Consortium at info@dsdm.org

2. Choosing DSDM as your Agile Approach

2.1 RAD and Agile - How It All Started

When DSDM was created in 1994, the world of solution delivery through projects was very different from how it is today. For example, the corporate world predominantly used a traditional (Waterfall) approach. Far too many of these projects were failing, for a variety of reasons, but mainly because projects were just too big, and too long, communication was poor and progress was measured in percentages, rather than deliverables. When projects did deliver, they often delivered late, and often delivered the wrong thing, due to lack of on-going business involvement and reliance on specifications which tried, and usually failed, to capture and fix detailed requirements right at the start.

To counter these problems, some projects had tried a completely different approach – Rapid Application Development (RAD) - with users of the solution working closely with developers to iteratively and incrementally build software applications, not based on a formalised specification, but on discussions, demonstrations and short feedback loops. This addressed many of the problems of the traditional approach but, in doing so, it introduced a whole new set of problems, particularly around the supportability and scalability of the solutions. RAD provided quick fixes but often its application adversely affected the quality of the solutions because the *disciplines* of analysis and design were thrown out with the up-front phases that used to contain them.

At that time, DSDM was launched to address the problems of the traditional approach (too slow, too big, not transparent enough, not enough on-going business involvement) as well as the problems introduced by RAD (focus only on speed and quick fixes, no focus on quality, no view of the big picture issues). DSDM achieved this by recognising that both approaches had strengths and areas for improvement, and that to be effective in all environments requires the ability to deal with wider context issues as well as the here and now. So DSDM brought together the best parts from a traditional approach (control and quality) and from RAD (good communication, business involvement, transparency).

The term "Agile" was first used in 2001 after a group of like-minded people, including Arie Van Bennekum from the DSDM Consortium, agreed to meet for a weekend in Snowbird, Utah. At this meeting, they acknowledged that they all shared common values and ways of working and agreed a formalised set of those values and 12 supporting principles that defined an Agile way of working. This new way of working is fundamentally different in style from the traditional Waterfall approach that dominated the world of IT projects at the time. From this meeting sprang the Manifesto for Agile Software Development, and a group called the Agile Alliance, which still exists today. In recent years, the use of Agile has grown significantly.

2.2 DSDM, Agile and the Agile Alliance

Ever since its launch in 1994, DSDM has been at the forefront of scalable Agile projects and solution delivery. DSDM is equally effective on small straightforward solutions or large complex corporate projects. DSDM has been used effectively on non-IT solutions and is not just about development of software. DSDM is often referred to as "mature Agile", since it grew up with a strong base in the corporate world of projects from 1994 and retains a strong project focus in the 21st century.

As a founder member of the Agile Alliance, DSDM has been at the heart of Agile since 2001. The philosophy and principles of DSDM helped shape the Manifesto for Agile Software Development, although DSDM takes the concept of Agile far wider than just software. The DSDM Agile Project Framework fully adopts the values laid out in the Manifesto.

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:

Individuals 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 in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Individuals and interactions **over processes and tools**

In an Agile project, great emphasis is placed on the team. Every individual in the team is expected to be ready, willing and able to play their part in the project, carrying out their role with competence and professionalism. Every member of the team is expected to work collaboratively with everybody else, using his or her knowledge, experience and judgement to shape a project outcome that best meets the need of the sponsoring business. Processes and tools play an important part in any project but much less emphasis is placed on these in an Agile environment. Agile processes need to be light touch and serve to guide and support rather than dictate what individuals and teams do and how they should do it. The assumption is that the team is best placed to understand what needs to be done and to work out the best way of doing it. DSDM provides appropriate light touch guidance on roles and responsibilities, whilst keeping the emphasis at all times on the people and the way they work together.

Working software **over comprehensive documentation**

The choice of words used here reflects the origins and primary focus of Agile – a focus solely on software delivery. However, changing a single word - "software" to "solution" - elevates this value from delivery of a software product

to encompass the broader context of business change projects and often this is the interpretation for DSDM. DSDM has been proven to work equally well for non-software projects.

The message behind this Agile value is to break the illusion of security and stability that comes from document-driven processes. Specification of every detail of requirements, solution design, plans etc. in documents that get 'signed off' by stakeholders before work is allowed to progress is both wasteful and ineffective. DSDM embraces the need for high-level versions of these artefacts in the early phases to frame development and delivery projects and to support governance. After the Foundations phase in a project, the framework employs collaborative techniques with active business engagement to ensure that the right solution is delivered. The framework also advocates light and timely documentation to support the solution in production beyond the end of the project.

Customer collaboration **over contract negotiation**
This Agile value encourages project teams and the sponsoring business to work collaboratively at all times. Typical commercial contracts assume that a traditional Waterfall process underpins development and 'a fixed price for a fixed specification' is the standard for project contracts. Agile projects emphasise collaboration, and therefore contracts need to reflect this.

A contract can be as formal as a document signed by those responsible for sponsoring the solution and by those delivering it, or as informal as a shared verbal agreement on what is to be delivered. Regardless of formality, it is important to ensure that, where created, all parties follow the principle of such documents or agreements being 'light touch' and 'guiding' rather than being 'detailed' and 'prescriptive'. By this definition, DSDM's Prioritised Requirements List may represent a contract, effectively defining the scope of a project. However, as it is at a high level, it requires customer collaboration with less formality to expand on the detail of requirements throughout the iterative development of the solution during the project lifecycle.

Responding to change **over following a plan**
This Agile value emphasises the fact that the world around a project is rarely frozen in time. Change occurs so quickly in the world of business now. that adopting an approach to building a solution which accommodates, or ideally embraces, change is likely to be the only way to a successful outcome. Change may also arise as a result of an emerging understanding of what is needed, what is valuable and even what is possible. The degree of change in detail that is typical of most projects makes creating detailed, long-term plans a waste of time. Where change is normal rather than the exception, the high-level 'light touch' and 'guiding' plans advocated by DSDM better meet the need of a flexible business.

And the very important final sentence
The DSDM Agile Project Framework specifically embraces the last sentence in the Agile Manifesto that clearly states, in the context of the values above

> *"That is, while there is value in the items on the right, we value the items on the left more."*

It is important not to ignore processes, tools, documentation, contracts and plans but instead to ensure that they are only created where they add value, and only to the level of detail that adds value. They should be created in a form relevant to and taking full advantage of the Agile values.

2.3 How Does DSDM Differ From More Traditional Approaches?

DSDM is a vendor-independent approach focused on helping people to work effectively together to achieve business goals. It can be used in any business, in any technical environment for any project.

A fundamental assumption of the DSDM approach is that nothing is built perfectly first time, but that as a rule of thumb 80% of the value of the solution can be delivered for 20% of the effort that it would take to produce the total solution (Pareto's Principle). A basic problem with less Agile approaches is the entirely unrealistic and unreasonable expectation that those responsible for specifying a solution can predict what all their requirements will be at some distant point in time and in exact detail. This problem is compounded by the fact that a new solution as it evolves is a stimulus for change, as understanding of the impact the solution will have on the target business grows.

In the traditional, sequential (or 'Waterfall') approach, the next step cannot be started until the current step is completed. In practice, a lot of time is wasted in each step aiming for perfection when actually an 80% solution would probably suffice. The additional effort to achieve 100% is justified on the basis that no step ever needs to be revisited if it was completed 'properly' first time around. In reality, considerable time is spent going back to 'completed' steps and unravelling the defects from work that has previously been accepted but was based on assumptions that turned out to be either false or which simply changed over time. The result of this potentially tortuous rework of detail is that projects are delivered late and over budget (if the rework is successful) or they fail to meet the business need (if the rework is avoided or rushed).

In DSDM, the iterative approach encourages detail to emerge over time; therefore, the current step needs to be completed in only enough detail to allow the project to move to the next step with any shortfall in detailed understanding being dealt with in a subsequent iteration of development. There is also a very strong likelihood that the business requirements will change over time, and that such change is most likely to happen at the detail level. This being the case, the effort spent on detailed up-front work is very likely to be wasted. In addition, solutions built using the DSDM approach address the current and imminent needs of the business rather than, for example, the traditional approach of attacking all the perceived possibilities.

The resulting solution is more likely to have a better fit with the true business needs, is easier to test and easier to integrate into existing and emerging business processes. The development cost of most solutions is only a small part of the total lifecycle costs; it therefore makes sense to build simpler solutions that are both fit for purpose on the day of delivery and easier to maintain and modify thereafter. This is preferable to trying to implement a more extensive solution that has been complicated and often compromised by failed attempts to predict future business needs.

2.4 How does DSDM Differ From Most Agile Approaches?

In addition to addressing many of the problems inherent in a traditional approach, DSDM also addresses many of the general concerns about Agile development. Specifically, DSDM requires basic foundations for the project to be agreed at an early stage. This allows businesses to understand the scope and fundamental characteristics of the proposed solution, and the way it will be created, before development starts. Clarifying and agreeing the foundations for the project from the combined perspectives of business, solution and management reduces the likelihood of nasty surprises on DSDM projects. In particular, for larger corporate organisations or organisations with a complex architecture and/or governance standards, agreeing the foundations early in the project is essential.

DSDM also describes a broader set of roles than most Agile approaches giving it a better fit with most corporate environments without compromising Agility.

2.5 How DSDM Addresses Key Project Problems

Managing any business change or developing any solution is rarely a simple task. Certain problems occur regularly whenever people from multiple disciplines work together on a project. DSDM is specifically designed to address many of these well-known problems. The following are seen as the key problems.

The Issue	How DSDM Helps
Ineffective communication	Poor communication is highlighted time after time as a major failing on projects. Establishing clear and concise communication between the different areas and levels of an organisation is not easy. DSDM provides a lot of guidance to strengthen communication and uses practices that encourage this to be visual and verbal wherever possible. DSDM's emphasis on human interaction (e.g. through the use of workshops), visualisation (e.g. through the use of modelling, prototyping and iterative development) and clearly defined roles is at the heart of excellent project communication. In particular, visualisation has proved to be a far more effective way of communicating than the use of large, textual documents, passed from one person to another, and sometimes used to apportion blame when an unworkable solution has been delivered.
Late delivery	Slippage of the promised completion date often causes much frustration, as well as having a detrimental impact on a business. DSDM sees this issue as one of the most important problems to address. The DSDM approach, and many of the practices it exploits, are focused on delivering on time. Being on time applies to short-term goals as well as to the project as a whole. If there is ever a need for compromise on a project, DSDM advocates that revising the scope of what is delivered should always be considered ahead of a extending the deadline. Under most circumstances the vast majority of the benefit of a solution can still be gained from a solution with the less important features left out.
The delivered solution does not meet the business needs	Another frustration arises when a solution is delivered which doesn't meet the expectations of the business. It may have features that don't do what the business really wanted it to do, or contain errors which prevent the deliverable from performing smoothly, or it simply might not be properly aligned with business processes. In DSDM, bringing the correct understanding of the needs of the business into the project is of paramount importance. To ensure that this is achieved business representatives are included as part of the Solution Development Team and DSDM encompasses practices which encourage collaboration and enable visual and verbal communication. Most importantly, DSDM teams are encouraged to embrace change, allowing them to deal with problems and opportunities that occur, to encompass new ideas that appear and to build the solution based on a deepening understanding of the solution detail.
Building the right thing – the business changing their mind	A frequent cry from those building the solution on a traditional project is that 'the users have changed their minds'. Far from being a problem, DSDM embraces change and believes that change often arises as the result of a deepening understanding or an unavoidable external event. DSDM capitalises on the greater depth of understanding and so ensures that the Deployed Solution meets the true business need. DSDM enables change through iterative development, with regular reviews to make sure what is being developed is what the business really needs. Requirements change is a natural result of a better understanding: DSDM expects it and plans for it.
Unused features	Research has highlighted that a relatively low percentage of features delivered as part of the overall solution developed using traditional approaches are actually used. This often happens because the business is encouraged to define all of its possible needs and wants at the outset of a project. By helping the business to prioritise its needs as understanding of the detail grows, DSDM keeps focus on what is important. This also avoids causing delays to a project by developing features that are never used.
Delayed or late Return on Investment (ROI)	Usually, the business value of a feature decreases over time and therefore delivering everything towards the end of a project may reduce the ROI. DSDM uses incremental delivery to get the most important and most valuable features released to the business as early as is practical. When appropriate, it can aggressively harness techniques such as vertical prototyping in order to deliver a subset of the total solution to the business very early and therefore to enable early return on investment.
Over-engineering ('Gold plating')	There is normally a diminishing return (on value) when trying to make a deliverable 'perfect'. Usually the highest business value can be derived by getting something that is 'good enough' into a window of opportunity for the business. Indeed, sometimes the entire window of opportunity may be missed and no value at all delivered. DSDM is a pragmatic approach which focuses on the real business need in order to dissuade a team from being tempted to add the final extras which the business could live without. Prioritisation ensures the whole team are clear about the relative importance of the work to be done and that low value 'polishing' of the solution does not impact deadlines and compromise ROI.
"Fragile" Agile	Many organisations have adopted Agile behaviours and approaches but have done so in a very casual and disorganised manner. In an attempt to reduce the burden of bureaucracy, they have gone to the other extreme and created a very ad hoc situation which is typified by poor discipline, little documentation and a general feeling of chaos. DSDM helps here by placing the right Agile concepts and constructs in a structured and controlled framework that enables a project to respond to change and stay in control, whilst still being fully Agile.
Waterfall dressed up as Agile	A common mistake made when transitioning to Agile is to use the iterative and incremental way of working but constraining it by applying an overall Waterfall project lifecycle. The most common example is where iterative and incremental timeboxed development follows on from traditional analysis and design steps in the waterfall and is followed by a traditional testing step. Whilst this may appear at the team level to be Agile and is probably more efficient than a big poorly controlled block of development activity, it fails to exploit the full potential of early delivery of real business value and does not mitigate the risks associated with inadequate business engagement. DSDM does just enough work up front to ensure clarity of objectives and to provide a foundation for solution development. This foundation is agreed before breaking the project down into Increments and within that to Timeboxes, ensuring the appropriate elements of detailed analysis, design, build and test at each level. Active engagement of business roles in the detail of development ensures the right solution evolves. This, in combination with the limited up-front work creates a truly Agile way of delivering benefits to the business.

"In general DSDM is very proactive in its nature with regard to these kinds of risks."

2.6 Why Choose DSDM As Your Agile Approach

There are a number of Agile approaches available, and although all support an iterative style of working with continuous business involvement, beyond that, the focus is different. Choosing an Agile approach that does not actually address all the needs of the business can introduce unnecessary risk into an organisation.

DSDM has a broader focus than most other Agile approaches in that it deals with projects rather than just the development and delivery of a product (typically software). The project context requires a focus on the wider business need and all aspects of the solution that evolves to meet that need. DSDM has a long track record of successful Agile project delivery in all types of corporate environments, and has proved to be fully scalable, working effectively in small simple businesses, large, complex organisations and in highly regulated environments. It also has been shown to be equally effective for both IT and non-IT projects, for example business change projects.

DSDM may also be used to supplement an existing in-house Agile approach, where this has proved to be lacking. For example, DSDM is often used to provide the full "project" focus to complement Scrum's team focussed product development process. The Agile Project Management and Scrum pocketbook provides guidance on this particular combination.

DSDM also takes a pragmatic approach, recognising that it often needs to work alongside existing standards and approaches. Examples of this are DSDM with Prince2, DSDM with ITIL, DSDM with formal quality processes, such as ISO or CMMI and DSDM with a PMO.

DSDM is not only about developing new solutions; projects to enhance existing solutions are also well suited to the DSDM approach.

The ethos of DSDM and the DSDM Consortium is to embrace and partner with the Agile community at large. We recognise and value the various Agile approaches and practices and believe that good Agile can be a single or blended approach, whichever is the right solution for your project environment. As the use of Agile increases, new ideas surface frequently, and this is why DSDM sees the need to evolve and embrace the wider Agile community for the greater good and to continually improve what is seen as best practice.

2.7 Summary of the Benefits of Using DSDM

Using iterative development, DSDM involves the solution's business stakeholders throughout the project lifecycle. This has many benefits, for example:

- The business is more likely to feel ownership of the solution as it evolves and, most importantly, as it transitions into live use
- Prioritisation will enable a project to be delivered on time whilst protecting the quality of what is being delivered
- The risk of building the wrong solution is greatly reduced
- The final solution is more likely to meet the real business need
- Deployment is more likely to go smoothly, because of the co-operation of all parties concerned throughout development

DSDM specifically addresses many of the problems which cause projects to struggle or to fail. For many organisations, having the ability to deliver working solutions consistently, on time and on budget, is seen as a major step forward. DSDM will provide this.

3. Philosophy and Fundamentals

3.1 Introduction

The DSDM philosophy is that

> *"best business value emerges when projects are aligned to clear business goals, deliver frequently and involve the collaboration of motivated and empowered people."*

This is achieved when all stakeholders:

- Understand and buy into the business vision and objectives
- Are empowered to make decisions within their area of expertise
- Collaborate to deliver a fit for purpose business solution
- Collaborate to deliver to agreed timescales in accordance with business priorities
- Accept that change is inevitable as the understanding of the solution grows over time

Stakeholders encompass everybody inside or outside the project who are involved in or affected by it.

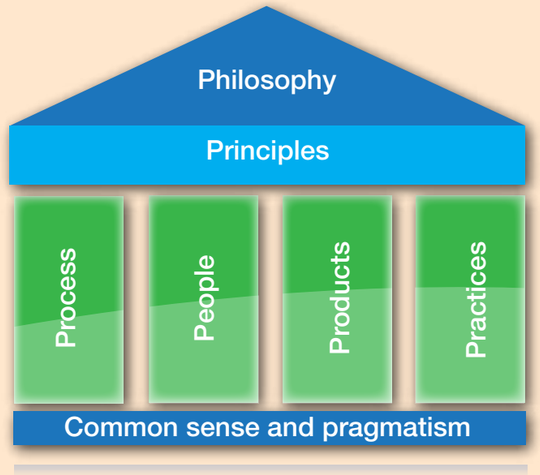


Figure 3a: The composition of DSDM

The DSDM philosophy is supported by a set of eight principles that build the mindset and behaviours necessary to bring the philosophy alive. The principles are themselves supported by people (with defined roles and responsibilities), an Agile process (enabling an iterative and incremental lifecycle to shape development and delivery), clearly defined products and recommended practices to help achieve the optimum results.

DSDM's approach and style has always been founded on an underlying ethos of common sense and pragmatism. It may be useful to clarify the meaning of these words:

- Common sense - "sound practical judgment independent of specialised knowledge or training; normal native intelligence."
- Pragmatism - "action or policy dictated by consideration of the immediate practical consequences rather than by theory or dogma."

This is the style of thinking that underpins "the way DSDM works". It is this flexibility of thinking that enables DSDM to avoid the dogmatic thinking that is sometimes encountered in the world of Agile. The ethos of common sense and pragmatism ensures that "individuals and interactions" continue to take precedence over "processes and tools".

3.2 Expected Project Outcomes using DSDM

Projects embracing the DSDM philosophy, adhering to the principles and following all the guidance associated with the people, process, products and practices will deliver the right solution at the right time for the right cost and will allow the sponsoring business to achieve the benefits that justified the project in the first instance.

It is, however, unrealistic to expect everything within the areas of the business, the solution and the management of the project, to align in such a way as to be perfect for the application of DSDM in its entirety on every project. A pragmatic phrase associated with DSDM since its inception states "While you can use all of DSDM some of the time, you can always use some of DSDM" and it is the fact that DSDM is a flexible framework that allows this to be true.

The following sections in this chapter, and the detail provided in subsequent chapters, assume circumstances which support the application of DSDM. Throughout the handbook, reference is also made to circumstances that are less than ideal. The chapter on tailoring DSDM deals explicitly with some of the most common challenges, using the experience of many DSDM practitioners over many years to show how these challenges have been addressed successfully using DSDM.

3.3 Understanding Project Variables

Projects have to balance conflicting demands, and the four most common demands are: time, cost, features and quality. Trying to fix all four at the outset of a project is unrealistic, as this would only work in a perfect world where the business need never changes, everything is fully and precisely understood in advance, and problems never happen. This desire to fix everything is the cause of many project failures, as the lack of sufficient contingency results in flawed decisions, and these usually affect the project towards the end.

For this reason, it is important at the start of a project to ask the question "If we hit a problem, what do we protect (fix) and what can we negotiate (vary) if necessary?"

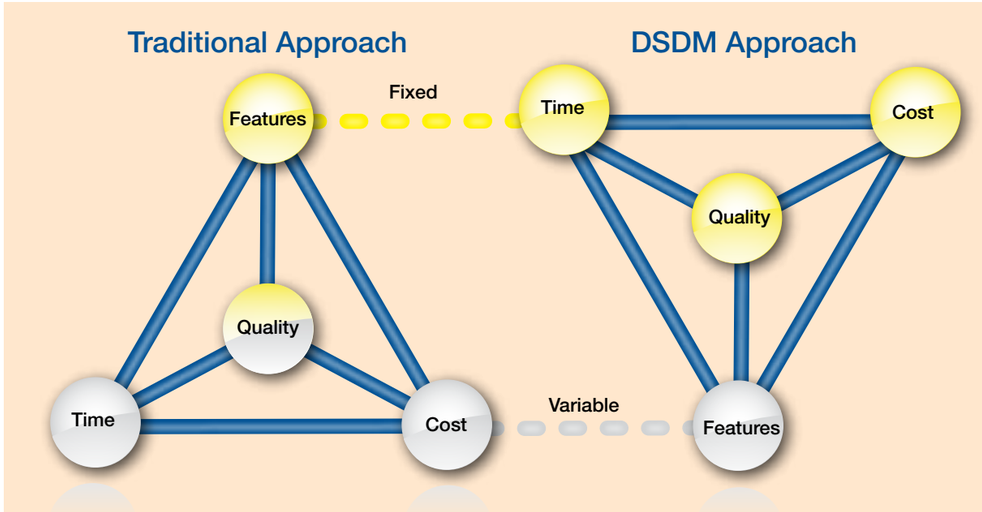


Figure 3b: Project variables – traditional and DSDM

In the traditional approach to managing a project (left-hand diagram), the features content of the solution is fixed whilst time and cost are subject to variation.

If the project goes off track, more resources are often added (which varies the cost) and/or the delivery date extended (which varies the time). However, adding resources to a late project often makes it even later and a missed deadline can be disastrous from a business perspective and often damages credibility. Under such pressure, quality often becomes a variable, as a result of introducing compromises which have not been thought through, by reducing essential quality control steps or by cutting back on testing.

DSDM's approach to managing the project (right-hand diagram) fixes time, cost and quality at the end of the Foundations phase while contingency is managed by varying the features (the requirements) to be delivered. As and when contingency is required, lower priority requirements are dropped or deferred, with full agreement of the business stakeholders in accordance with MoSCoW priorities.

A DSDM project will always deliver a viable solution, on time and on cost (on budget), as long as the practices of MoSCoW and timeboxing are followed. Delivery of a Minimum Usable SubseT of requirements is guaranteed as a worst case scenario. However, in all but the most extreme circumstances, the expectation is to deliver far more than the bare minimum.

The level of quality for each project depends on the needs of the individual project. Not every project requires "highest quality", but it does need to achieve the agreed quality level for this project. One of the ways DSDM fixes quality on a project is by agreeing acceptance criteria for individual requirements before development commences. The iterative and incremental approach to development ensures that the more important requirements are built to the agreed level of quality. Only once this has been achieved does development start on the less important requirements. Incremental delivery of the Evolving Solution ensures that, on the day that the solution is deployed into live use, quality is at the level expected and previously agreed.

3.4 DSDM Delivers the Right Solution at the Right Time

3.4.1 Delivery at the right time

DSDM ensures solutions are delivered at the right time by breaking the project down into focused, fixed duration Project Increments, and within these into one or more timeboxes also of fixed duration and lasting typically, two to four weeks. In order to protect the agreed end date of an Increment or Timebox, agreement is in place from the start that, if necessary, low priority requirements will be de-scoped.

3.4.2 Delivering the right solution

Using DSDM, the right solution is delivered to the business because:

- The project team and other significant stakeholders remain focused on the business needs
- All people involved with the project work collaboratively to achieve that outcome
- DSDM harnesses the knowledge, experience and creativity of teams to understand the business problem or opportunity, and then to work together to build the optimum solution
- A limited amount of early work ensures firm foundations to support the solution as it grows
- Work is prioritised according to business need and the ability of the business to accommodate changes in the agreed timescale
- An iterative and incremental approach to development and delivery of the solution assures alignment with business need
- Quality is never allowed to become a variable

3.5 Summary

To enable the philosophy of driving out best business value through projects aligned to clear business goals, frequent delivery and collaboration of motivated and empowered people, DSDM offers eight principles, supported by definition of and guidance on people, products, process and practices. All of this guidance needs to be applied with common sense and pragmatism; adapting to the project's environment and context, while preserving the ethos of DSDM presented here.

4. Principles

4.1 Introduction to the DSDM Principles

The eight principles of DSDM support DSDM's philosophy that

> *"best business value emerges when projects are aligned to clear business goals, deliver frequently and involve the collaboration of motivated and empowered people".*

They also bring the Agile values to life by guiding the team in the attitude it must take and the mindset it must adopt in order to deliver consistently whilst still remaining flexible.

Compromising any of the following principles undermines the philosophy of DSDM and introduces risk to the successful outcome of the project. If a team doesn't follow all of these principles then it won't get the full benefit of the approach. The collective value of DSDM's principles enables organisations to deliver best value business solutions collaboratively.

The eight DSDM principles are:


1. Focus on the business need


2. Deliver on time


3. Collaborate


4. Never compromise quality


5. Build incrementally from firm foundations


6. Develop iteratively


7. Communicate continuously and clearly


8. Demonstrate control

4.2 Principle 1 - Focus on the Business Need

Every decision taken during a project should be viewed in the light of the overriding project goal - to deliver what the business needs to be delivered, when it needs to be delivered.

It is important to remember that a project is a means to an end, not an end in itself.

In order to fulfil this principle, DSDM teams will:

- Understand the true business priorities
- Establish a valid business case
- Ensure continuous business sponsorship and commitment
- Guarantee delivery of the Minimum Usable SubseT (this is explained in detail in the section on MoScoW prioritisation)

Specific business roles in DSDM, in conjunction with the business products created in the Foundations phase, and key practices such as timeboxing and MoSCoW prioritisation, enable DSDM teams to fulfil this principle.

4.3 Principle 2 - Deliver on Time

Delivering a solution on time is a very desirable outcome for a project and is quite often the single most important success factor. Late delivery can often undermine the very rationale for a project, especially where market opportunities or legal deadlines are involved. Even for projects without a need for a fixed end date, on time delivery of intermediate or contributing products is still the best way to demonstrate control over evolution of the solution.

In order to fulfil this principle, DSDM teams need to:

- Timebox the work
- Focus on business priorities
- Always hit deadlines
- Build confidence through predictable delivery

Combining the DSDM practices of timeboxing and MoSCoW prioritisation enables DSDM teams to protect deadlines whilst flexing the features, and to build a reputation for timely and predictable delivery. The ability to deliver on time and to meet the prioritised expectations of the business in the short term – the timebox – forms the basis of control over the longer-term delivery of the project through timely delivery of Increments.

4.4 Principle 3 – Collaborate

Teams that work in a spirit of active cooperation and commitment will always outperform groups of individuals working only in loose association. Collaboration encourages increased understanding, greater speed and shared ownership, which enable teams to perform at a level that exceeds the sum of their parts.

In order to fulfil this principle, DSDM teams need to:

- Involve the right stakeholders, at the right time, throughout the project
- Encourage pro-active involvement from the business representatives
- Ensure that all members of the team are empowered to take decisions on behalf of those they represent
- Build a one-team culture

DSDM's Business Visionary, Business Ambassador and Business Advisor roles bring the appropriate subject matter experts into the project so they can contribute to the solution. The Solution Development Team brings together business and technical roles in a single team. This one-team culture is fostered by the Business Analyst helping to facilitate business agreement on the requirements and the Team Leader taking responsibility for facilitating a high level of collaboration between all Solution Development Team members. Facilitated workshops enable stakeholders to share their knowledge effectively with other members of the project team.

4.5 Principle 4 - Never Compromise Quality

In DSDM, the level of quality to be delivered should be agreed at the start. All work should be aimed at achieving that level of quality - no more and no less. A solution has to be 'good enough'. If the business agrees that the features in the Minimum Usable SubseT meet the agreed acceptance criteria, then the solution should be 'good enough' to use effectively.

In order to fulfil this principle, DSDM teams need to:

- Agree the level of quality from the outset, before development starts
- Ensure that quality does not become a variable
- Test early, test continuously and test to the appropriate level

- Build in quality by constant review
- Design and document appropriately

Ensuring testing is properly integrated into the Iterative Development process, with regular reviews throughout the project lifecycle, helps the DSDM team to build a quality solution. The review and quality control products created as the project proceeds help demonstrate that the quality of the solution is meeting the expected standard.

Using DSDM, everything is tested as early as possible. MoSCoW prioritisation and timeboxing are used to ensure that testing is appropriate and undertaken without introducing unnecessary risks. In an IT project, the use of test-driven development techniques can also significantly improve the quality of the solution by ensuring that the acceptability of the solution is understood before development starts.

4.6 Principle 5 - Build Incrementally from Firm Foundations

One of the key differentiators for DSDM within the Agile space is the concept of establishing firm foundations for the project before committing to significant development. DSDM advocates first understanding the scope of the business problem to be solved and the proposed solution, but not in such detail that the project becomes paralysed by overly detailed analysis of requirements.

Once firm foundations for development have been established, DSDM advocates incremental delivery of the solution in order to deliver real business benefit as early as is practical. Incremental delivery encourages stakeholder confidence, offering a source of feedback for use in subsequent Timeboxes and may lead to the early realisation of business benefit.

In order to fulfil this principle, DSDM teams need to:

- Carry-out appropriate analysis and enough design up front (EDUF) to create strong foundations
- Formally re-assess priorities and informally re-assess ongoing project viability with each delivered Increment

DSDM teams implement this principle through the appropriate application of a project lifecycle, which delivers a solid base of knowledge during Feasibility and Foundations phases before building the solution incrementally during the Evolutionary Development phase.

4.7 Principle 6 - Develop Iteratively

DSDM uses a combination of Iterative Development, frequent demonstrations and comprehensive review to encourage timely feedback. Embracing change as part of this evolutionary process allows the team to converge on an accurate business solution. The concept of iteration is at the heart of everything developed as part of the DSDM approach. It is very rare that anything is created perfectly first time and it is important to recognise that projects operate within a changing world.

In order to fulfil this principle, DSDM teams need to:

- Build business feedback into each iteration
- Recognise that most detail should emerge later rather than sooner
- Embrace change – the right solution will not evolve without it
- Use iterative development to encourage creativity, experimentation and learning

Change is inevitable; DSDM allows for change and harnesses its benefits.

Within the constraints of time and cost, change is actively encouraged in order to evolve the most appropriate solution. DSDM uses iteration and constant review to make sure that what is being developed is what the business really needs. Cycles of feedback should form part of the process for evolving all project deliverables e.g. all plans and documentation.

4.8 Principle 7 - Communicate Continuously and Clearly

Poor communication is often cited as the biggest single cause of project failure. DSDM practices are specifically designed to improve communication effectiveness for both teams and individuals.

In order to fulfil this principle, DSDM teams need to:

- Encourage informal, face-to-face communication at all levels
- Run daily team stand-up sessions
- Use Workshops, with a facilitator where appropriate
- Use visual communication practices such as Modelling and Prototyping
- Demonstrate the Evolving Solution early and often
- Keep documentation lean and timely
- Manage the expectations of the stakeholder at all levels throughout the project
- Always aim for honesty and transparency in all communication

DSDM emphasises the value of human interaction through Stand-ups (see Chapter 13 - Timeboxing), Workshops, clearly defined roles and active business involvement.

Modelling and Prototyping make early instances of the solution available for scrutiny. These practices are far more effective than the use of large textual documents, which are sometimes written for reasons other than achieving the business objectives of the project.

4.9 Principle 8 - Demonstrate Control

It is essential to be in control of a project at all times and to be able to demonstrate that this is the case. This can only be achieved by reference to a plan for the work being done, which is clearly aligned with agreed business objectives. It is also vital to ensure transparency of all work being performed by the team.

In order to fulfil this principle, DSDM teams, especially the Project Manager and Team Leader, need to:

- Make plans and progress visible to all
- Measure progress through focus on delivery of products rather than completed activities
- Manage proactively
- Evaluate continuing project viability based on the business objectives
- Use an appropriate level of formality for tracking and reporting

The use of well-defined Timeboxes, with constant review points, and the preparation of the Management Foundations and Timebox Plans, are designed to assist the Project Manager and the rest of the project team to follow this principle.

4.10 Summary

The eight principles help direct and shape the attitude and mindset of a DSDM team. Compromising any of the principles undermines DSDM's philosophy, as together they deliver a collective value that outweighs their individual benefits.

5. Preparing for Success

5.1 Introduction – Instrumental Success Factors (ISFs)

The following factors are seen as instrumental for positioning DSDM projects for a successful outcome. Where these factors cannot be met, they represent a significant risk to the DSDM approach. Therefore, it is important to identify these risks early and consider how they could be mitigated. Many projects successfully use DSDM whilst still identifying that some of these factors will not be in place. These projects recognise that a DSDM approach still offers reduced risk, as well as a much higher probability of success, compared with adopting a more traditional approach that often fails to deliver the expected outcomes.

5.2 Embracing the DSDM Approach

It is important that all project stakeholders and participants understand and accept the DSDM project approach. As well as embracing the DSDM philosophy that *best business value emerges when projects are aligned to clear business goals, deliver frequently and involve the collaboration of motivated and empowered people*, this also includes the concept that in order to deliver the right thing at the right time and to handle change dynamically, the project may deliver less than 100% of the possible solution.

For example:

A DSDM project had delivered on the agreed deadline all the Musts and the Shoulds, but was unable to deliver every Could Have requirement. So the Minimum Usable SubseT and the expected business case had been delivered. However the Sponsor flagged the project as a failure, since it had not delivered 100% of the requirements, even though a number of changes to detail raised by the business had been incorporated. It is clear that in this case the Sponsor had not accepted (or had forgotten) a critical aspect of the DSDM approach.

5.3 Effective Solution Development Team

People are at the heart of successful DSDM projects and the Solution Development Team is instrumental in ensuring the development of the right solution. Building an effective team for successful delivery focuses on four elements:

- Empowerment
- Stability
- Skills
- Size

5.3.1 Appropriate empowerment of the Solution Development Team

Each role within the Solution Development Team should be empowered to make decisions based on their expertise, and the team as a whole empowered to make decisions within the boundaries agreed during foundations.

Within the business roles, the senior business management (Business Sponsor, Business Visionary) must agree to delegate day-to-day decision-making to the Business Ambassador(s) in the Solution Development Team. If this is not possible, they will need to participate in the team themselves (i.e. in this circumstance, the Business Ambassador role may need to be taken by a more senior person from the business). Without business empowerment, team progress will slow down while waiting for decisions being made elsewhere and made to a different timeframe.

For example:

For a DSDM project, committed to on-time delivery, the delay caused by putting a decision on hold whilst waiting for the next scheduled committee meeting for approval poses a significant problem and a risk to the agreed deadline.

The Business Ambassador(s) should be empowered to make the day-to-day business decisions without referral to higher authorities outside the team.

For example:

If a Business Ambassador does not have the authority to make simple day-to-day decisions within a timebox (such as agreeing a Could Have can be dropped), this will cause unnecessary delays while people outside the team have to be contacted and briefed in order to "approve" detailed decisions. This will impact the commitment to deliver on time.

Similarly, other roles within the Solution Development Team should be empowered to make detailed, day-to-day decisions about how the solution should be built and tested.

It is important to understand that the concept of empowerment does not give all Solution Development Team members complete freedom to do whatever they want, whenever they want. In reality, empowerment is always within agreed boundaries of decision-making and typically these boundaries are agreed as part of Foundations.

For example:

The Solution Development Team may agree to change the detail of a requirement or the way in which it is implemented, but they cannot change the overall scope of the project.

Where a decision falls outside the agreed boundaries of the team empowerment, this would need to be formally escalated. However, this is the exception and the majority of day-to-day decision-making should be within the remit of the Solution Development Team.

The level of empowerment to be given to the business roles, especially the Business Ambassador, should be clarified in the early phases of the project and validated throughout.

5.3.2 Solution Development Team stability

The Solution Development Team brings together business and technical knowledge throughout the iterative development process. In addition, as the solution evolves dynamically, this places great emphasis on conversations between team members, rather than relying on documents as the primary focus of communication. This means that a DSDM project will be put at serious risk if team members are swapped in and out. Business and Technical Advisors may be called in on an intermittent as needed basis, as long as the core team remains stable.

Some organisations routinely swap people in and out of teams. However each change has a significant impact both on the team shared knowledge base and on the team dynamics. Much of the information on a DSDM project is gained through face-to-face discussions and group understanding, with far less reliance on detailed signed-off static documents. Where team changes have to be made, for preference this should happen at the end of a Project Increment..

5.3.3 Solution Development Team skills

Progress is significantly enhanced when the Solution Development Team(s) contain skilled people, both in terms of business knowledge and technical expertise. This does not mean that every team member needs to be a multi-skilled expert; it means that all the core skills for the project must be present within the Solution Development Team as a whole. However team members need good communication skills and the willingness to work with others, if a team with diverse skills is to function as a coherent unit.

For example:

Where all technical members of a team comprise people with very high technical skills but with very low team/people/communication skills, this poses a significant risk to DSDM (and any Agile approach), since DSDM is a team-based approach relying on good inter-team communication. To be effective, the communication needs to happen spontaneously.

5.3.4 Solution Development Team size.

DSDM teams rely on informal communication as their first choice. For this to be effective, DSDM suggests that the optimum Solution Development Team size is seven +/- two people. (The Solution Development Team comprises the roles of Team Leader, Business Ambassador, Business Analyst, Solution Developer and Solution Tester.)

At this level, the team can communicate with one another with:

- Minimum formality
- Minimum management overhead
- Minimum risk
- Maximum benefit and ownership

Although smaller and larger team sizes have proven to be effective in a DSDM project environment, both have specific risks associated with them which need to be addressed.

- Where the Solution Development Team is very small, (e.g. three or four people) there is a risk associated with delivering what has been promised from a Timebox. If one person is absent from the team for any reason (for example, through illness), this may represent a very significant percentage of the capacity to complete the work.
- Where the Solution Development Team is greater than nine, the communications become more complex, daily stand-ups take longer (which may impact productivity) and some of the team communication may need to be managed more formally.

One project may have a number of Solution Development Team's. Where the team size is going to be greater than DSDM's recommended team size, splitting into a number of smaller teams may be a better option, although this in itself will introduce an overhead to manage the various teams. The options, benefits and risks should be assessed to ensure the most suitable choice for an individual project.

For example:

A DSDM project had a Solution Development Team of 30 (ignoring advice not to work this way). Each daily stand-up took over 1 hour (@ 2 minutes per person), so each day this stand-up equated to 30 hours of project time used. And the value of communication in such a large group was extremely limited.

5.4 Business Engagement - Active and On-going

In order for DSDM projects to be successful, it is vital that the business is actively engaged and commits the necessary amount of time at all levels, and that this commitment is maintained throughout the project.

Ensuring active and on-going business engagement relies on three elements:

- Commitment of business time throughout
- Day-to-day collaboration involving business roles in the Iterative Development process
- A supportive commercial (e.g. contractual) relationship (where appropriate)

5.4.1 Commitment of business time throughout

The business commitment and agreed participation is vital to successful DSDM projects, since these roles provide the business direction to the project. In the early phases, business engagement and business time is needed to provide the vision, the high-level requirements, the business priorities and the business case for the work to be done. In the later phases, the business roles provide the lowest-level detail and prioritisation of the requirements during Timeboxes, as well as testing and on-going acceptance of the Evolving Solution.

The level of business commitment for the project should be quantified, discussed and agreed in the early phases. Without this commitment, the success of the DSDM approach may be limited, especially where initial enthusiasm for a different way of working has passed and other business commitments start to draw on Business Ambassador or Business Advisor time.

It is also important to recognise that the key business people who have the necessary level of knowledge and empowerment are often also those who are very busy and have limited time. This is especially true for the Business Ambassador role. Successful DSDM projects rely on getting business people with good knowledge, and not just those who happen to have some free time. This makes it even more important that calls on the business time are managed effectively to ensure the maximum value is gained from the time available, and that the business understand clearly how much time is expected from them.

For example:

A DSDM project needed access to a highly respected (and very busy) Business Ambassador. It was agreed that in order to provide the necessary level of support and detail to the project, he would allocate 7 hours per week to the project, and would also be available to answer phone calls from 9 – 9.30 each day. To help cover this commitment, the Business Sponsor agreed some additional administrative support for the Business Ambassador, so he was not expected to do 7 days work in a 5-day week.

5.4.2 Active Involvement of the business roles

The best communication occurs if the Solution Development Team (which includes the Business Ambassador) is co-located in their own dedicated environment, free from daily interruptions, although this ideal scenario is not always possible. For successful DSDM projects, contact with the business roles must be on-going and frequent throughout the project. Where the whole Solution Development Team is not co-located, planning becomes even more important, since when someone is not sitting nearby, the ease of immediate communication is missing.

For example:

Where the Solution Developers are in a different time zone from the Business Ambassador(s), there will be periods where it is not possible simply to pick up the phone; this potential delay presents a risk to the DSDM approach which needs to be addressed. One approach is to plan for a daily communication session during the shared working time. This could include the daily stand-up, some form of Q&A session and where possible, a show-and-tell.

5.4.3 A supportive commercial relationship

Where the supplier and the customer are from different organisations and development is covered by formal contract, or where Solution Developers are from the same organisation but working within a service level agreement, the relationship must accommodate the evolution of the solution's requirements without imposing onerous change management overheads.

5.5 Iterative Development, Integrated Testing and Incremental Delivery

On all DSDM projects, ensuring testing is fully embedded as part of the iterative and incremental development approach is key both to the reduction of project risk and to the success of the project. Ensuring that individual elements of the solution are technically fit for purpose and meet the business need, builds confidence in the direction and the quality of the Evolving Solution. Linking Solution Increments together and testing them in order to prove they are still behaving as expected delivers an even higher level of confidence. On all types of project, testing early and often mitigates the risk of discovering deeply embedded faults that may take significant time and effort to rectify, too late to take proper corrective action.

Ensuring that testing is an integral part of development also opens up options for incremental deployment of the solution. An organisation amenable to incremental delivery of solutions into live use will benefit from early return on investment and a reduction in deployment risk (compared with the big-bang, large drop of a 100% final solution at the end of a project). In addition, releasing a partial solution allows the business to adopt the solution in manageable chunks and allows them to provide feedback on what is actually being delivered. It also ensures the Evolving Solution builds on the firm foundation of a previous release, meaning that the Solution Development Team are always building from a position of confidence.

The concept of incremental delivery also carries down to individual Timeboxes, where each Timebox ideally delivers a complete potentially-deployable increment of the solution. Where the Business Ambassador can accept requirements/user stories as "done" at the end of a Timebox, this provides tangible reassurance that a Solution Increment really meets business expectations, even though there may not be enough at that point in time to physically deploy it. The value of this reassurance cannot be underestimated.

5.6 Transparency

DSDM is all about building confidence in the Evolving Solution, and in this way reducing the risks of the unknown or the invisible. The value of building confidence through incremental delivery of business value, ideally demonstrating fully completed requirements/user stories (built, tested and accepted by the business) at the end of each Timebox, cannot be underestimated. If this cannot be achieved then an alternative demonstration of progress may be needed.

> *A real IT example:*
>
> *A DSDM project was incrementally delivering an ecommerce solution with new or revised features being demonstrated at the end of each timebox. Part way through the project, the Technical Coordinator advised the team that the compiler settings needed to be changed. This identified inefficiencies in the way the code was written that were illustrated by several pages of warning messages. One member of the team agreed to work to resolve the issues in the worst affected and most complex code module and decided to demonstrate progress by running the compiler on original and refactored versions of the code. The results of his efforts earned him a round of applause when he demonstrated that he had eliminated over 95% of the warnings. No demonstrable business value but still an obvious improvement.*

Demonstrations of the Evolving Solution provide physical, objective and unquestionable proof of progress (compared with a progress report showing a subjective % complete). In practice, ensuring the business see elements of the actual solution during Timebox demonstrations (Show and Tells), combined with proof that business feedback is being used to converge on an accurate solution, often means that the need for formal Waterfall-style progress reports becomes less relevant. Team Boards (see Chapter 14.2.5) can also be very useful for providing clear and up-to-date information on the current state of work for the Timebox, the Project Increment and even the whole project, where appropriate.

Making activity and progress transparent through demonstrations (rather than relying on written reports that are intended to describe such progress) ensures that the business is always fully aware of the true state of the project. This helps decisions to be made earlier, when there are more options available.

5.7 The Project Approach Questionnaire - Assessing Options and Risks

In order to set projects up for the best chance of success, it is important to take a realistic look at the working environment, the people and the relationships and assess the risks. This may then lead to some tailoring of the DSDM approach - as a framework, DSDM is designed to be tailored to allow a close fit to a variety of environments. Further guidance on tailoring DSDM can be found later in this handbook. The starting point for this assessment is DSDM's Project Approach Questionnaire (PAQ). This simple checklist assesses a number of key areas for DSDM success, and starts to identify potential risks to DSDM success which need to be addressed. Normally, the PAQ is first completed (by the Project Manager in conjunction with the project-level roles of Business Sponsor, Business Visionary and Technical Coordinator) during Feasibility. It should be re-assessed towards the end of Foundations with the project-level roles and the Solution Development Team. By this point many of the risks may have already reduced, but other risks may have surfaced. The information from the PAQ is used to inform the approach to be taken by the project for development and delivery and to drive active management of the project risks.

The full detail of the PAQ can be found in Chapter 17 (Tailoring the DSDM Approach) and a template of the PAQ can be found in Appendix B. A more detailed template with high-level guidance notes can be downloaded from www.dsdm.org. Some organisations find it helpful to add supplementary questions, based on their specific needs/risks.

5.8 Summary

Understanding and assessing the factors that are instrumental for success in the early phases of a DSDM project can help significantly in addressing and mitigating potential risks to the success of the project. Having a common understanding of what needs to be in place is a good starting point for any project and working towards achieving the best starting position for a DSDM project increases the likelihood of a successful project.

6. Process

6.1 Overview

In line with the DSDM philosophy that

> *"the best business value emerges when projects are aligned to clear business goals, deliver frequently and involve the collaboration of motivated and empowered people"*

the DSDM approach to development and delivery is both iterative and incremental, with the most important business needs typically being addressed early while less important features are delivered later. The iterative nature of DSDM enables business representatives to see the solution as it evolves, to provide feedback on it and to request changes throughout the development of the solution.

Unlike most Agile approaches, DSDM integrates project management and product development into a single process. For many organisations, DSDM is all that is needed, although some gain value from integrating DSDM with other methods e.g. project management methods, such as PRINCE2™ and PMI, or software engineering practices, from eXtreme Programming (XP).

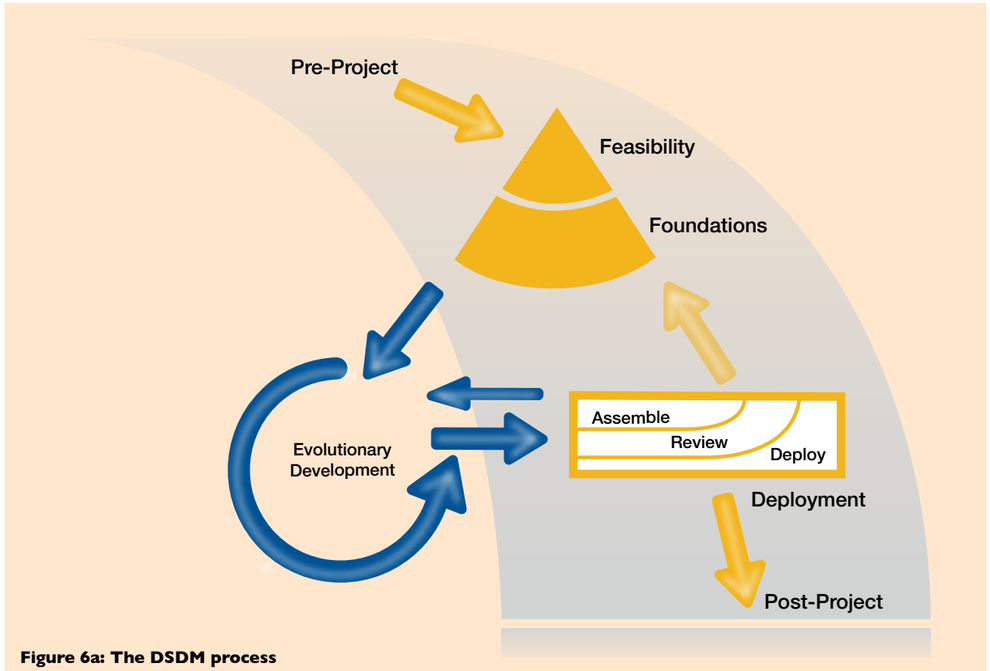


Figure 6a: The DSDM process

The DSDM process model comprises a framework which shows the DSDM phases and how they relate to one another. This process model is then used by each project to derive their lifecycle.

The project process, as shown in the figure above, has four main phases: Feasibility, Foundations, Evolutionary Development and Deployment. These are preceded by the Pre-Project phase and followed by the Post-Project phase, giving six phases in total. All phases are formally defined later in this chapter in terms of their purpose and what needs to be in place before they can be successful.

6.2 Pre-Project Phase

In line with the DSDM Philosophy

> *"that best business value emerges when projects are aligned to clear business goals"*

the Pre-Project phase ensures that only the right projects are started, and that they are set up correctly, based on a clearly defined objective.

6.3 Feasibility Phase

The Feasibility phase is intended primarily to establish whether the proposed project is likely to be feasible from a technical perspective and whether it appears cost-effective from a business perspective. The effort associated with Feasibility should be just enough to decide whether further investigation is justified, or whether the project should be stopped now, as it is unlikely to be viable.

6.4 Foundations Phase

The Foundations phase takes the preliminary investigation from Feasibility to the next level. It is intended to establish a fundamental (but not detailed) understanding of the business rationale for the project, the potential solution that will be created by the project, and how development and delivery of the solution will be managed. By intentionally avoiding low levels of detail, the Foundations phase should last no longer than a few weeks - even for large and complex projects. The detail associated with requirements, and how they should be met as part of the solution, is intentionally left until the Evolutionary Development phase of the project.

It may sometimes be necessary to revisit Foundations after a Deployment phase. The decision to revisit Foundations may be planned in from the start of the project; for example, on a project where the business environment is sufficiently dynamic that the Foundations are expected to encounter significant change during the life of the project. Alternatively, the decision to revisit Foundations may be taken after a Deployment has produced an unexpected outcome.

Returning to the Foundations phase to re-affirm and, where necessary, refine the foundations of the project normally takes significantly less time than establishing them in the first place and may be as short as a single Workshop.

For smaller, simpler projects, the Feasibility and Foundations phases can often be merged into a single phase.

The aim of Foundations is to understand the scope of work, how it will be carried out, by whom, when and where. The Foundations phase also determines the project lifecycle by agreeing how the DSDM process will be applied to the specific needs of this project.

6.5 Evolutionary Development Phase

Building on the firm foundations that have been established for the project, the purpose of the Evolutionary Development phase is to evolve the solution.

The Evolutionary Development phase requires the Solution Development Team(s) to apply practices such as Iterative Development, timeboxing, and MoSCoW prioritisation, together with Modelling and Facilitated Workshops, to converge over time on an accurate solution that meets the business need and is also built in the right way from a technical viewpoint.

Working within Timeboxes, the Solution Development Team create Solution Increments, iteratively exploring the low-level detail of the requirements and testing continuously as they move forward.

6.6 Deployment Phase

The objective of the Deployment phase is to bring a baseline of the Evolving Solution into operational use. The release that is deployed may be the final solution, or a subset of the final solution.

Some examples of what can be deployed are:

- Business change - introducing a new way of working into a factory (deploying a business change as a single release)
- The early deployment of a corporate intranet, providing a limited number of features, with more features to be provided later (deploying the first release of many)
- A complex product - e.g. the launch of a new mobile phone, bringing together parts of the solution from multiple projects run in different locations (deploying a new product as a single release)

The Deployment phase comprises three main activities: Assemble, Review and Deploy. In addition, after the last release, the project is formally closed.

6.6.1 Assemble

Before a physical deployment, there are usually activities that take place to ensure that what is being delivered is coherent. This may also include bringing together any relevant supporting information. Assemble encompasses the work to "bring together" what is to be released.

On a small simple project, the work involved during Assemble may be minimal. On larger more complex projects or programmes where multiple projects are feeding into a single release, the amount of work to assemble a number of Solution Increments into a single release could be significant e.g. combining a new business process, a schedule of training, user guides and a new IT solution.

6.6.2 Review

Once all the elements of a release have been assembled, in most circumstances there will be some form of "approval to deploy". This will be based on a final review of the solution before it goes into operational use - to ensure the proposed release meets the appropriate standards and is complete enough to be viable. In a simple environment, this can be very informal – a basic checklist - but in a more complex environment, it may be as formal as a go/no-go checkpoint workshop.

At this point, the team also carries out a retrospective for the Project Increment, focusing on ways of working and potential areas for improvement.

Information from both the retrospective and the formal review of the product help shape plans for future increments and can be used to facilitate learning across projects within a portfolio.

6.6.3 Deploy

Once approval has been given, Deploy is the physical act of putting what has been assembled (the release) into operational use. It includes any technical work, such as transfer of the solution into the live (production) environment, but also the enactment of any plans for business change.

6.6.4 Closing the project

After the final Deployment, the project is formally closed. At this point, the whole team hold a retrospective to review the overall project performance, both from the technical and/or process perspective and from the business perspective.

6.6.5 Deployment and complexity

A release may encompass one or more Solution Increments and can also span one or more projects. This means that the Deployment phase may be a simple or a complex activity. How deployment is done varies from organisation to organisation, and from project to project. For many organisations, decisions about how deployment is handled are imposed by the organisation itself, and are not negotiable by an individual project.

There are two common scenarios for deployment:

- Deployment phase is under the control of the project; or
- The project has control of Assemble and Review but not the final Deploy activity

6.7 Post-Project Phase

After the final Deployment for a project, the Post-Project phase checks how well the expected business benefits have been met. Although it may be possible to highlight immediate benefits, most benefits will accrue over a pre-defined period of live use of the solution.

The Post-Project phase produces one or more Benefits Assessments for these realised benefits in relation to the business case. Benefits may be assessed for individual releases (in which case the assessment of benefit should start before the Post-Project phase is reached), for the whole project or may be omitted completely, depending on the needs of the organisation.

6.8 The Lifecycle in Practice

Whilst there is a clear progression of phases from Pre-Project to Post-Project in the process diagram above, there are also arrows indicating a return path within the process, specifically the arrows from Deployment to Foundations and from Deployment to Evolutionary Development. The process shows the framework and the options available, and then each project derives their lifecycle from this process. The lifecycle for a project is determined by factors such as the number of intended Project Increments and other external influences, such as the stability of the business environment and endurance of Foundations decisions. The lifecycle for the project is defined and agreed as part of the Foundations phase.

6.9 Configuring DSDM for Scalability and Formality

DSDM recognises the real value of Agility in terms of project productivity and solution quality while acknowledging and accepting the necessary constraints that often exist when working in a corporate environment. Such constraints may include financial governance, architecture and/or infrastructure strategies, regulatory governance, vendor agreements and third party support considerations.

The DSDM process can be configured and calibrated to cater for a range of projects: small projects with light governance, larger projects which need stronger governance. Typically this is achieved by configuring a lifecycle appropriate for a specific project and determining an appropriate level of formality with which the DSDM products are defined, created and approved.

With regards to scaling, the project organisation can easily be refined to support multiple teams, with key roles acting as directors and coordinators across the teams. To support a more complex project structure, products such as the Solution Architecture Definition, Development Approach Definition, Management Approach Definition and the Delivery Plan and Timebox Review Records can be made more elaborate and more formal than would be appropriate for smaller projects.

6.10 Summary

DSDM provides an iterative and incremental process, with a total of six lifecycle phases. Each phase has a specific purpose, together with a number of defined products intended to support the evolution of the solution and the smooth running of the project. The DSDM Agile Project Framework is designed to work effectively with projects of varying size and complexity. Through the tailoring of its various products, DSDM ensures control is demonstrated to a level of formality appropriate to the organisation, thereby running a project so that all the benefits of Agile are achieved without compromising effective project governance.

7. Roles and Responsibilities

7.1 Introduction

People working together effectively are the foundation of any successful project. DSDM recognises this and assigns clear roles and responsibilities to each person in a project, representing the business interests, the solution/technical interests, the management interests and the process interests. Everyone involved in a DSDM project works very closely together in order to break down potential communication barriers.

The best solutions emerge from self-organising, empowered teams. However, these teams, and the people within them, must actively take on the responsibility for their empowerment within the boundaries that have been agreed. At the same time, it is important they:

- Respect each other's knowledge, experience, skills and opinions
- Take personal responsibility for their work and the dependence of the other team members on them
- Have the courage to challenge ways of working, to improve their team collaboration and working processes

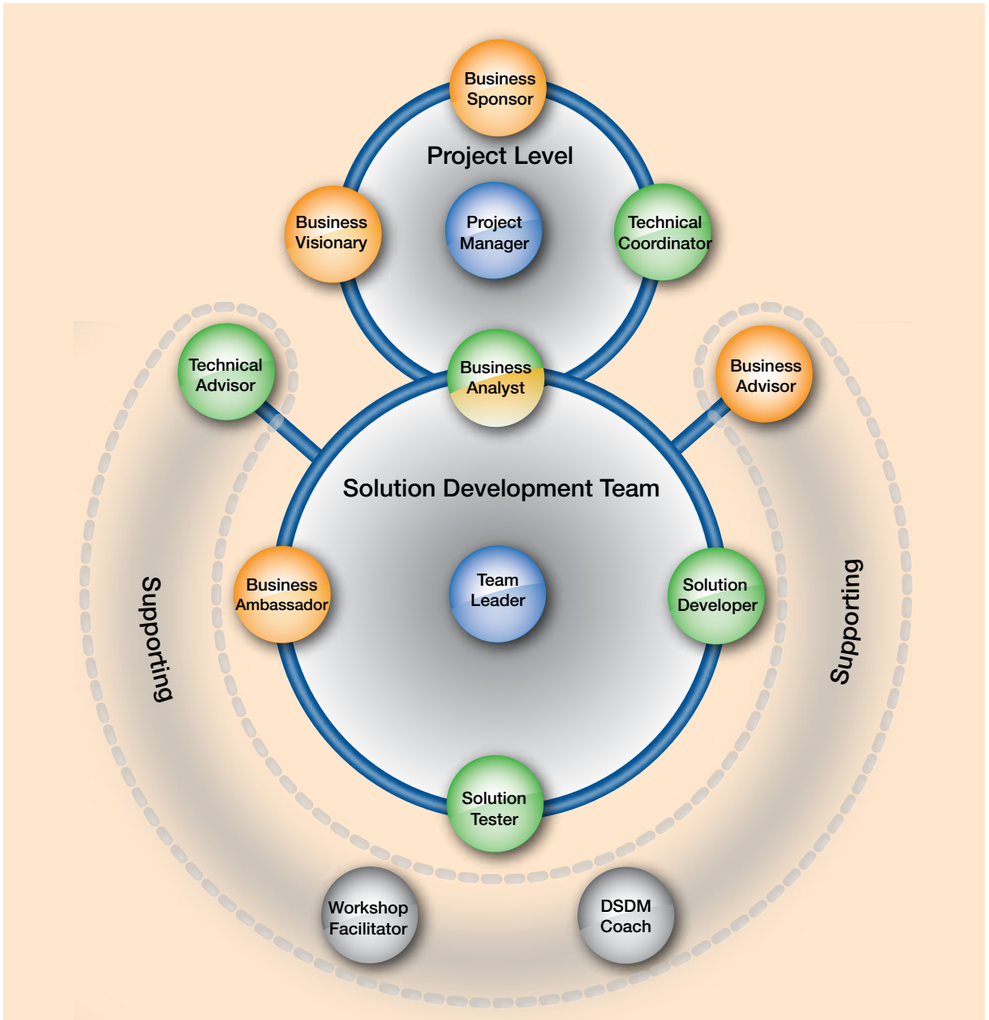


Figure 7a – The DSDM team model

7.2 The DSDM Team Model Explained

7.2.1 Role colour scheme - to represent areas of interest

The colour scheme in figure 7a is as follows:

- **Orange** - Business interests, roles representing the business view
  - Typically taken by business personnel, e.g. Business Ambassador providing day-to-day business direction, Business Visionary providing high-level direction and a view of the future.
- **Green** - Solution/technical interests, roles representing the solution/technical view
  - Contributing to the technical development of the solution, e.g. Solution Developers creating the solution, Technical Coordinator providing technical leadership and direction.
- **Blue** - Management interests, roles representing the management/leadership view
  - Facilitating the management/leadership aspects of the project, e.g. Project Manager and Team Leader following the DSDM process and managing/leading a DSDM project (using Agile leadership competencies).
- **Grey** - Process interests, roles representing the process view
  - Facilitating the process aspects of the project, e.g. Workshop Facilitator managing the workshop process, DSDM Coach embedding the DSDM framework
- **Mix of two colours** – A role that straddles two separate areas of interest, e.g. Business Analyst, has both a business and a solution/technical focus

7.2.2 Role Categories

7.2.2.1 Project-level roles

In figure 7a, the project-level roles (Business Sponsor, Business Visionary, Technical Coordinator, Project Manager and Business Analyst) are the directors, managers and coordinators of the work for the project, where necessary. They may be part of a project board or steering committee for the project and, collectively, have authority to direct the project. They are responsible for the governance of the project, liaising with governance authorities outside of the project. The Business Sponsor provides the overall strategic direction and controls the funding/budget for the project. The Business Visionary and the Technical Coordinator hold the business and technical visions, respectively, for the project. The Project Manager ensures that project funds are used effectively to create the envisaged solution within the agreed timescale.

The Business Analyst is intentionally positioned as part of the project level as well as part of the Solution Development Team. This allows the Business Analyst to, for example, help the business to formulate the Business Case, and also to be involved in assisting the business in defining their requirements during feasibility and foundations, sometimes before the full Solution Development Team. is assigned. The role then continues in supporting the Solution Development Team alongside the project-level roles, as the more detailed requirements emerge.

All roles at the project level need to adopt the facilitative, empowering leadership style which allows Agile teams to learn as they go and reflect, adapt and enhance process. They need to ensure the freedom of the Solution Development Team. to do the job, getting to an end point by its own means, within an empowerment framework for the team.

The project-level roles:

- Build projects around motivated individuals
- Trust the teams, confident that everyone will work to the best of their ability
- Give the teams the environment and support they need

7.2.2.2 Solution Development Team roles

The Solution Development Team roles are Business Ambassador, Solution Developer, Solution Tester, Business Analyst and Team Leader. These roles form the "engine room" of the project. They shape and build the solution and are collectively responsible for its day-to-day development and for assuring its fitness for business purpose. There may be one or more Solution Development Teams within a project. Each team will include all Solution Development Team roles and cover all their responsibilities.

The membership of each Solution Development Team should be stable throughout a project, however, in the worst case, each Solution Development Team should remain stable for a Project Increment. Each member of the Solution

Development Team is an empowered individual who takes personal ownership for their area of responsibility and represents the interests of their peers.

7.2.2.3 Supporting roles

The supporting roles (Business Advisors, Technical Advisors, Workshop Facilitator and DSDM Coach) provide assistance and guidance to the project on an ad hoc basis throughout the lifecycle. The Advisor roles may be filled by one or more subject matter experts, as necessary. The Advisor roles are not the empowered decision-makers – that is the responsibility of the roles within the Solution Development Team – but they advise the Solution Development Team in areas where specialist expertise is needed (e.g. legal and compliance matters, technical knowledge, business-specific rules and regulations). The supporting roles engage with the project as and when necessary. For example, a Business or Technical Advisor will be actively involved during Foundations and then for the particular Timeboxes where their expertise is needed to properly shape the Evolving Solution.

7.2.3 Levels of engagement

All DSDM roles need to be appropriately engaged in the project sufficiently to fulfil the responsibilities of their role. Project-level roles need to be engaged sufficiently to ensure that the ongoing work of the project remains aligned to the business need, is generating a solution to the agreed quality and continues to be viable in terms of the Business Case. Project-level roles therefore need to be engaged in high-level reviews and planning sessions, and perhaps in more detailed sessions where key issues and strategic decisions need their input. Their involvement is not normally needed or expected day to day but is more likely to be focussed around the beginning and end of Timeboxes and perhaps at key review points within them. Solution Development Team roles need to be actively engaged in the project on a day-to-day basis working at the detailed level; shaping, building, reviewing and testing the Solution Increment delivered at the end of each Timebox. All roles must attend the Daily Stand-up in order to maintain a common understanding of progress and any issues and, as a self-organising team, agree detailed plans and actions needed to meet their delivery commitments. Continuous, open, honest communication and day-to-day collaboration are the key to making good progress with transparency of progress and work being important in demonstrating control. Where project-level roles do engage at a lower level of detail, it is important that they do so as observers and leaders and the owners of issues rather than as managers of the team or the work being undertaken.

7.2.4 Fulfilling the roles

One DSDM role does not necessarily mean one person. One person may take on one role, or one person may cover two or more roles. One role may be split between two or more people. However, where a role is split between individuals, it is vital that these individuals communicate and collaborate closely.

> *For example:*
>
> *On a large IT project, the Technical Coordinator's responsibilities may be allocated to more than one person, e.g. the System Designer/Architect, the Networks Manager, the Infrastructure Manager etc.*
>
> *On a branding project, Solution Developer responsibilities may be split, with one Solution Developer focusing on logo design, another on key marketing messages.*

Conversely, in smaller projects, one person often performs more than one role.

> *For example:*
>
> *One person may carry out the responsibilities both of the Project Manager and the Team Leader.*
>
> *However, some roles are typically only fulfilled by one person, whatever the size of the project, e.g. there should only be one Business Visionary (rather than a group of visionaries) and one Business Sponsor. (Although it is also often true that one person fulfils both the Business Sponsor and the Business Visionary roles).*

Issues such as geographical constraints or staff availability can affect the creation of the ideal project team, but it is strongly recommended that all the roles are considered and that their individual responsibilities are all understood and accepted as appropriate. The role definitions can be used as the basis for personal terms of reference for a project.

7.3 Business Sponsor

This role is the most senior project-level business role. The Business Sponsor is the project champion who is committed to the project, to the proposed solution and the approach to delivering it. The Business Sponsor is specifically responsible for the Business Case and project budget throughout (however formally or informally this may be expressed).

The Business Sponsor must hold a sufficiently high position in the organisation to be able to resolve business issues and make financial decisions. This role has a crucial responsibility to ensure and enable fast progress throughout the project.

The Business Sponsor should be committed, supportive and available for the duration of the project, providing a clear escalation route. On smaller projects, the Business Sponsor role will always be fulfilled by a single person. However, on larger projects or in complex organisations, the Business Sponsor's financial responsibilities may be fulfilled by a higher authority such as an investment board or an executive committee. In this circumstance, DSDM expects the business to agree a specific person to "front" the role. This ensures the project deals with a single ultimate decision-maker and a single ultimate escalation point, and is protected from a lack of clarity through differing views about the project.

7.3.1 Responsibilities

- Owning the Business Case for the project
- Ensuring ongoing viability of the project in line with the Business Case
- Holding the budget for the project
- Ensuring that funds and other resources are made available as needed
- Ensuring the decision-making process for escalated project issues is effective and rapid
- Responding rapidly to escalated issues and being the ultimate point for resolution of conflict within the project
- Empowering the business roles within the project, to appropriate levels, within their responsibilities
- Keeping themselves informed of progress and issues, e.g. by attending demonstrations at the end of Timeboxes and asking questions of other roles who are more actively engaged

7.4 Business Visionary

This is a senior project-level business role that should be held by a single individual, since a project needs a single clear vision to avoid confusion and misdirection. More actively involved than the Business Sponsor, the Business Visionary is responsible for interpreting the needs of the Business Sponsor, communicating these to the team and, where appropriate, ensuring they are properly represented in the Business Case. The Business Visionary remains involved throughout the project, providing the team with strategic direction and ensuring that the solution delivered will enable the benefits described in the Business Case to be achieved. At the end of the project, the Business Visionary will own the Deployed Solution and will be responsible for the realisation of any benefits associated with it.

7.4.1 Responsibilities

- Defining the business vision for the project
- Communicating and promoting the business vision to all interested and/or impacted parties
- Monitoring progress of the project in line with the business vision
- Owning the wider implications of any business change from an organisational perspective
- Contributing to **key** requirements, design and review sessions, particularly where aspects of the solution being considered address key elements of the business vision

- Identifying and owning business-based risk
- Defining, and approving changes to, the high-level requirements in the Prioritised Requirements List, i.e. any change that affects the baselined scope or significantly alters the balance of priorities
- Ensuring collaboration across stakeholder business areas within the scope of the project
- Ensuring business resources are available to the project as needed
- Promoting the translation of the business vision into working practice, i.e. ensuring full business adoption of the solution created by the project
- Empowering the business roles within the Solution Development Team, to appropriate levels, within their responsibilities
- Where the Solution Development Team cannot agree, acting as an arbiter of business differences related to the business need and the way this is addressed in the Evolving Solution

7.5 Technical Coordinator

As the project's technical authority, the Technical Coordinator ensures that the solution/technical roles work in a consistent way, that the project is technically coherent and meets the desired technical standards. This role provides the glue that holds the technical aspects of the project together while advising on technical decisions and innovation.

The Technical Coordinator performs the same function from a technical perspective as the Business Visionary does from a business perspective.

7.5.1 Responsibilities

- Agreeing and controlling the technical architecture
- Determining the technical environments
- Advising on and coordinating each team's technical activities
- Identifying and owning architectural and other technically based risks
- Advising on the achievability of non-functional requirements
- Working with the Business Analyst to evaluate the technical options and decide the best way to turn the high-level business requirements into a technical solution
- Advising on and coordinating each team's approach to estimating, to reflect technical best practice and current technical understanding
- Promoting appropriate standards of technical best practice
- Controlling the technical configuration of the solution
- Approving the solution as technically fit for purpose prior to deployment
- Managing technical aspects of the transition of the solution into live use
- Empowering the technical roles within the Solution Development Team to appropriate levels within their responsibilities
- Acting as the final arbiter of technical differences between Solution Development Team members

7.6 Project Manager

As well as providing high-level Agile-style leadership to the Solution Development Team, the role is focused on managing the working environment in which the solution is evolving. The Project Manager also coordinates all aspects of management of the project at a high level but, in line with the DSDM concept of empowerment, the Project Manager is expected to leave the detailed planning of the actual delivery of the product(s) to the members of the Solution Development Team. Managing an empowered team requires a facilitative style rather than a "command and control" style.

Although the Project Manager role is focused on getting the project delivered, appropriate sourcing of the role will depend on the skills and knowledge required and on the project itself; the Project Manager may come from the business, or may come from the solution/technical side. For some projects, especially formal contractual projects being delivered by external suppliers, there may be two Project Managers, one from the business (the customer) and one from the solution/technical side (the supplier).

It is usual that the Project Manager takes responsibility throughout the duration of the project. This must include both business and technical delivery aspects of the project, from Foundations (if not Feasibility) through to Deployment.

7.6.1 Responsibilities

- Ensuring effective and timely communication and provision of information to project governance authorities (Business Sponsor, project board, steering committee etc.) and stakeholders not actively engaged in the project with the agreed and appropriate level of frequency and formality
- Performing high-level project planning and scheduling, but not detailed Timebox planning or task planning
- Collaborating with the Solution Development Team and other appropriate stakeholders to create and agree the Delivery Plan (the schedule of Project Increments and the Timeboxes within them)
- Monitoring progress against the baselined Delivery Plan
- Managing risk and any issues as they arise, collaborating with senior business or technical roles as required to resolve them
- Motivating and ensuring empowerment of the teams to meet their objectives
- Monitoring and ensuring appropriate involvement and communication between required members of the multi-disciplinary Solution Development Team
- Handling problems escalated from the Solution Development Team
- Providing help and guidance to the Solution Development Team where difficult situations arise
- Attending stand-up meetings, as appropriate, to keep a current understanding of the team's progress and issues, and to flag up to the team, where necessary, any important external issues that that team need to be aware of

7.7 Business Analyst

The Business Analyst is both active in supporting the project-level roles and fully integrated with the Solution Development Team. The Business Analyst facilitates the relationship between the business and technical roles, ensuring accurate and appropriate decisions are made on the Evolving Solution on a day-to-day basis. The Business Analyst ensures that the business needs are properly modelled and analysed and are correctly reflected in the guidance the team needs to generate the solution.

Active involvement of the business users in the process of evolving the solution is vital to the success of a DSDM project. Therefore it is important to ensure that the Business Analyst does not become an intermediary between the Solution Development Team members but, instead, supports and facilitates the communication between them.

7.7.1 Responsibilities

- Assisting the Business Visionary in the formulation and promotion of the business vision, as appropriate
- Modelling the organisation's current and future state in the area of the solution and identifying opportunities, risks and impacts
- Working with the Business Visionary and the Solution Development Team to formulate and communicate solution options
- Working with the project-level roles in formulating the Business Case and organising Benefits Assessments
- Supporting and facilitating unambiguous and timely communication between business and technical participants in the project
- Ensuring the requirements defined are of good quality and are analysed and managed appropriately

- Managing development, distribution and baseline approval of all communication related to business requirements and their interpretation, with particular focus on ensuring the prioritised requirements list is kept up to date, as the detail expands and evolves
- Ensuring that the business and organisational implications of day-to-day evolution of the solution are properly modelled and thought through
- Ensuring the impact of business decisions is reviewed in the context of the project
- Ensuring the business and technical components of the solution collectively provide a cohesive whole for the business
- Ensuring the non-functional requirements are achievable and subsequently met
- Taking responsibility for tracking business requirements through to business acceptance
- Liaising with the Business Visionary in organising support for the solution through implementation into live use

7.8 Team Leader

The Team Leader ideally acts as the servant-leader for the Solution Development Team and ensures that it functions as a whole and meets its objectives. The Team Leader works with the team to plan and coordinate all aspects of product delivery at the detailed level. This is a leadership role rather than a management role and the person holding it will ideally be elected by his or her peers as the best person to lead them through a particular stage of the project. It is therefore likely that they will also perform another Solution Development Team role (e.g. Business Analyst, Business Ambassador, Solution Developer or Solution Tester) in addition to their team leadership responsibilities. It is also feasible that the person carrying out the Team Leader role could be different from one Timebox to another, for example where they have a different focus.

7.8.1 Responsibilities

- Facilitating the team focus on the on-time delivery of agreed products
- Encouraging full participation of team members within their defined roles, responsibilities and empowerment
- Ensuring that the Iterative Development process is properly focused and controlled
- Ensuring that all testing and review activity is properly scheduled and carried out
- Managing risks and issues at the Timebox level, escalating to the Project Manager, Business Visionary or Technical Coordinator as required
- Monitoring progress on a day-to-day basis for all team activities
- Facilitating communication of team progress with the Project Manager
- Facilitating the daily stand-ups, ensuring they are timely, focussed and brief
- Facilitating reviews and retrospectives with the team

7.9 Business Ambassador

The Business Ambassador is the key representative of the business needs within the Solution Development Team and, as such, they need to have the desire, authority, responsibility and knowledge to fulfil the role.

During Foundations, the Business Ambassador has significant input into the creation and prioritisation of requirements. Once the requirements have been agreed and baselined (by the end of Foundations), the Business Ambassador then provides the day-to-day detail of the requirements during timeboxed development. This is either based on their own knowledge and experience, or drawing on the experience of the Business Advisors.

During the Evolutionary Development phase of the project, the Business Ambassador is the main decision-maker on behalf of the business. For this reason the Business Ambassador needs to be someone who is respected by their business peers and who has sufficient seniority, empowerment and credibility to make decisions on behalf of the business, in terms of ensuring the Evolving Solution is fit for business purpose. It is also important that the person fulfilling this role has the confidence to recognise where their own knowledge is insufficient and to bring in Business Advisors to support them.

Typically the Business Ambassador role is someone who is already busy. For this reason they must be able to commit the appropriate (and agreed) amount of time throughout Timebox development to help guide the Evolving Solution in the right direction to meet the business needs. For some projects, this may require a full-time commitment as the only way to meet the deadline. However this is unusual and actually introduces a risk that the Business Ambassador may become unaware of events occurring in the business. For most projects, the Business Ambassador commitment is a part-time one, at a level agreed during Foundations. But it is also important that where an Ambassador is committing time to the project, some of their normal workload can be delegated, so that all their work (day-to-day business and DSDM project) can be achieved in a normal working week. It is important that the amount of commitment expected is openly discussed and agreed at a workable level.

7.9.1 Responsibilities

- Contributing to **all** requirements, design and review sessions
- Providing the business perspective for all day-to-day solution development decisions
- Providing the detail of business scenarios to help define and test the solution
- Communicating with other users, involving them as needed and getting their agreement
- Providing day-to-day assurance that the solution is evolving correctly
- Organising and controlling business acceptance testing of the solution
- Taking responsibility for the creation of the business user and support documentation for the ultimate solution (this responsibility may be delegated, for example to a specialist such as a Technical Author, but the ultimate responsibility remains with the Business Ambassador)
- Ensuring business participants in the Deployed Solution are properly trained and supported

7.10 Solution Developer

The Solution Developer collaborates with the other Solution Development Team roles to interpret business requirements and translate them into a Solution Increment that meets functional and non-functional needs. A person assuming a Solution Developer role needs to be appropriately empowered by the Technical Coordinator to make day-to-day decisions in their area of expertise. They should ideally be allocated full-time to the project they are working on. Where they are not full-time, the project ought to be their first priority. If this cannot be achieved, significant risk is introduced with regard to timeboxing. This risk needs to be managed proactively by the Project Manager.

7.10.1 Responsibilities

- Working with all other Solution Development Team roles to iteratively develop:
  - The Solution Increment
  - Models required for the properly controlled development of the solution
  - Models and documents as required for the purpose of supporting the Deployed Solution in live use
- Testing the output of their own work prior to independent testing
- Agreeing and adhering to technical constraints
- Adhering to the organisation's technical implementation standards and best practice
- Participating in any quality assurance work required to ensure the delivered products are truly fit for purpose
- Recording (and later interpreting) the detail of any
  - Changes to the detailed requirements
  - Changes to the interpretation of requirements which result in re-work within the solution
  - Information likely to impact on the ongoing evolution of the solution

7.11 Solution Tester

The Solution Tester is an empowered Solution Development Team role, fully integrated with the team and performing testing throughout the project in accordance with the agreed strategy.

7.11.1 Responsibilities

- Working with business roles to define test scenarios and test cases for the Evolving Solution
- Carrying out all types of technical testing of the solution as a whole
- Liaising with the Business Analyst and Business Ambassador to help clarify acceptance criteria for requirements
- Creating test products as appropriate, e.g. test cases, test plans and test logs
- Reporting the results of testing activities to the Technical Coordinator for quality assurance purposes
- Keeping the Team Leader informed of the results of testing activities
- Assisting the Business Ambassador(s) and Business Advisor(s) so that they can plan and carry out their tests well enough to ensure that the important areas are covered

7.12 Business Advisor

Often a peer of the Business Ambassador, the Business Advisor is called upon to provide specific, and often specialist, input to solution development or solution testing - a business subject matter expert. The Business Advisor will normally be an intended user or beneficiary of the solution or may be a representative of a focus group. However they may, for example, simply provide legal or regulatory advice with which the solution must comply.

7.12.1 Responsibilities

Based on the specialism for which the Business Advisor has been engaged:

- Providing specialist input into relevant:
  - Requirements, design and review activities
  - Day-to-day project decisions
  - Business scenarios to help define and test the solution
- Providing specialist advice on, or help with:
  - Developing business user and support documentation for the ultimate solution
  - Deployment of the solution releases into the business, as appropriate

7.13 Technical Advisor

The Technical Advisor supports the team by providing specific, and often specialist, technical input to the project, often from the perspective of those responsible for operational change management, operational support, ongoing maintenance of the solution, etc.

7.13.1 Responsibilities

The Technical Advisor supports the Solution Development Team through the provision of detailed, and often specialist, technical input and advice with regards to:

- Requirements, design and review sessions
- The operational perspective for day-to-day decisions
- Operational or support scenarios to help define and test the solution
- Assurance that the solution is evolving correctly
- Operational acceptance testing
- Development of technical support documentation
- Training of technical operations and support staff
- Incremental Deployment of the solution releases, as appropriate

7.14 Workshop Facilitator

The Workshop Facilitator is responsible for managing the workshop process and is the catalyst for preparation and communication. The Facilitator is responsible for organising and facilitating a session that allows the participants to achieve the workshop objective.

The Workshop Facilitator should be independent of the outcome to be achieved in the workshop.

7.14.1 Responsibilities

Before each workshop:

- Agreeing the scope of the workshop with the workshop owner (the person who wants the workshop to take place)
- Planning the workshop, including agreement of empowerment and the decision-making process
- Familiarisation with the subject area of the workshop, if necessary
- Engaging with participants prior to the workshop to:
  - Confirm their suitability as a participant (in terms of knowledge, state of empowerment and their need to be at the workshop)
  - Ensure their full understanding of the workshop objectives
  - Understand any major areas of interest and concern in the subject area
  - Encourage completion of any required preparation work

During each workshop:

- Facilitating the workshop to meet its objectives

At the conclusion of each workshop

- Reviewing the workshop outcome against its objectives

After each workshop

- Ensuring the workshop results are distributed to participants and other agreed stakeholders, as necessary.

7.15 DSDM Coach

Where a team has limited experience of using DSDM, the role of the DSDM Coach is key to helping team members to get the most out of the approach, within the context and constraints of the wider organisation in which they work. The DSDM Coach should ideally be certified as a DSDM Coach to ensure that their competence to fulfil this role has been independently validated.

As with any method of working in any context, the approach cannot be followed blindly. If there is something in the project environment that will inhibit the effectiveness of a particular DSDM technique, then it is vital that the potential problem is addressed. Typically, there are two ways of addressing such a problem: the first is to influence the environment to allow the technique to be effective; the second is to adapt or substitute the technique. Either way, an expert in DSDM - the DSDM Coach - will have the knowledge and experience to help.

7.15.1 Responsibilities

- Providing detailed knowledge and experience of DSDM
- Tailoring the DSDM process to suit the individual needs of the project and the environment in which the project is operating

- Helping the team use DSDM practices and helping those outside the team appreciate the DSDM philosophy and values
- Helping the team work in the collaborative and cooperative way typical of DSDM and all Agile approaches
- Building DSDM capability within the teams at all levels

7.16 Summary

DSDM identifies roles in two dimensions – categories and interests.

Roles are grouped into three categories:

- Project roles
- Solution Development Team roles
- Supporting roles

Within a DSDM project, the different interests are represented using colours:

- Orange represents the business interests
  - This covers the Business Sponsor, Business Visionary, Business Ambassador and Business Advisor roles
- Green represents the solution/technical interests
  - This covers the Technical Coordinator, Solution Developer, Solution Tester and Technical Advisor roles
- Blue represents management interests
  - This covers the Project Manager and Team Leader roles
- Grey represents process interests
  - This covers the Workshop Facilitator and DSDM Coach roles

The Business Analyst role is coloured a mix of orange and green since this role often straddles the boundary between business and solution/technical interests.

On a DSDM project, one role may be fulfilled by several people, or one person may fulfil several roles.

8. Products

8.1 Introduction to DSDM Products

The DSDM Agile Project Framework describes a set of products to be considered as the project proceeds. These products describe the solution itself (the main deliverable of the project) and anything created to help with the process of evolving it, and anything that is required to help with project governance and control.

Not all products are required for every project and the formality associated with each product will vary from project to project and from organisation to organisation. The formality of the products is influenced by factors such as contractual relationships, corporate standards and governance needs.

DSDM identifies two distinct types of product:

- *Evolutionary products* evolve over time. They typically, but not always, span a number of project phases and may be baselined more than once during that time.
- *Milestone products* are created in a phase and typically fulfil a specific purpose within that phase as a checkpoint or to facilitate governance processes.

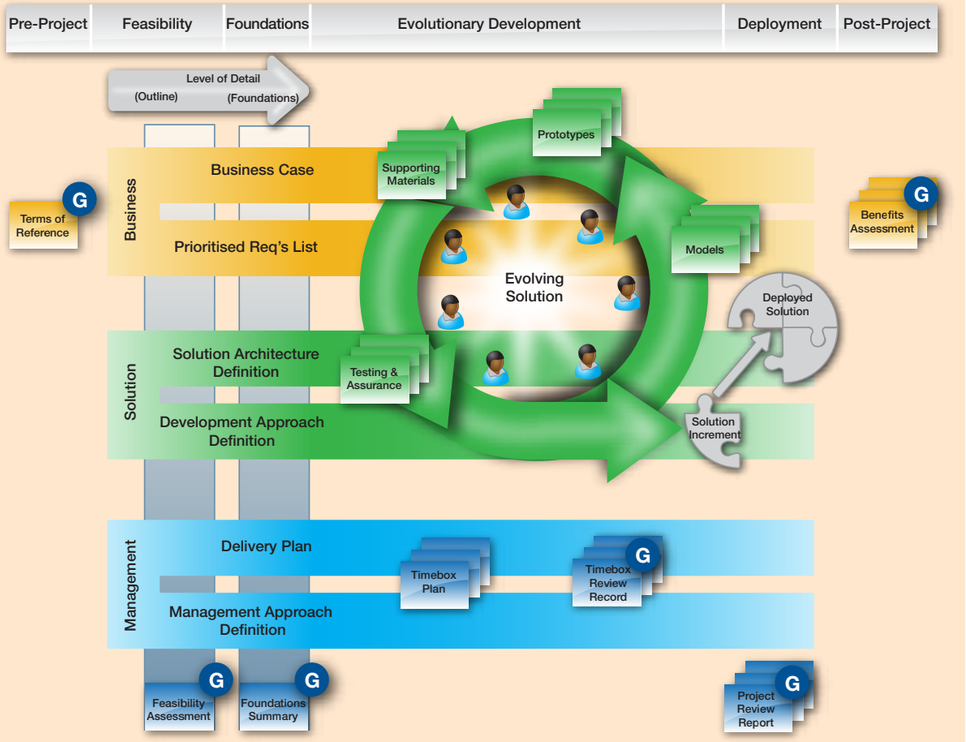


Figure 8a: DSDM products

The products, and where they feature in the project lifecycle, are shown in the diagram above. Orange products are business-focused, green products all contribute to the solution being created by the project and blue products cover project management/control interests. Several of the products - those marked with G - may also play a part in governance processes such as approval gateways, and may be used to demonstrate compliance of the solution with corporate and regulatory standards where this is required.

8.2 The DSDM Products

8.2.1 Terms of Reference

Description

The Terms of Reference is a milestone product. It is a high-level definition of the over-arching business driver for, and top-level objectives of, the project. The primary aim of the Terms of Reference is to scope and justify the Feasibility phase. It is identified as a governance product because it may be used for purposes such as prioritisation of a project within a portfolio.

Roles and responsibilities

	Role	Rationale
Produced by	Anybody	Anybody can have an idea for a project
Produced for	Project Governance Authority	To check alignment with strategic goals and help prioritise within a portfolio
	Business Analyst Technical Coordinator	To ensure, by reference, objectives and proposed solutions emerging during Foundations phase are appropriately aligned
Approved by	Business Sponsor	Person with budget for the Feasibility investigation

8.2.2 Business Case

Description

The Business Case is an evolutionary product. It provides a vision and a justification for the project from a business perspective. The business vision describes a changed business as it is expected to be, incrementally and at the end of the project. The justification for the project is typically based on an investment appraisal determining whether the value of the solution to be delivered by the project warrants the cost to produce, support and maintain it into the future, all within an acceptable level of risk.

Baselines of the Business Case are typically created first as an outline by the end of Feasibility, then as a basis for approval of development by the end of Foundations. It is formally reviewed at the end of each Project Increment in order to determine whether further work is justified.

Roles and responsibilities

	Role	Rationale
Produced by	Business Analyst	Skills and experience with Business Case production and working collaboratively with the senior business and technical roles
Produced for	Project Governance Authority	Approval for project to proceed and to help prioritise the project within a portfolio
	The entire project team	Everybody involved needs to understand what is needed and why
Approved by	Business Sponsor	Responsible and accountable for return on investment

8.2.3 Prioritised Requirements List

Description

The Prioritised Requirement List (PRL) is an evolutionary product. It describes, at a high level, the requirements that the project needs to address and indicates their priority with respect to meeting the objectives of the project and the needs of the business. Consideration of requirements begins in Feasibility and a baseline of the PRL demarcates the scope of the project as at the end of Foundations. After that point, further change will happen naturally in terms of depth, as a result of emergence of detail. Change to the breadth (adding, removing or significantly changing high-level

requirements) needs to be formally controlled in order to ensure ongoing alignment with the business vision for the project and to keep control of the scope.

Roles and responsibilities

	Role	Rationale
Produced by	Business Analyst	Skills and experience with eliciting and defining requirements
Produced for	The entire project team	Everybody involved needs to understand the requirements
Approved by	Business Visionary	Responsible for ensuring that requirements align with business vision

8.2.4 Solution Architecture Definition

Description

The Solution Architecture Definition is an evolutionary product. It provides a high-level design framework for the solution. It is intended to cover both business and technical aspects of the solution to a level of detail that makes the scope of the solution clear but does not constrain evolutionary development.

Roles and responsibilities

	Role	Rationale
Produced by	Business Analyst	Responsible for overall design of business process and organisation change
	Technical Coordinator	Responsible for overall design and integrity of technical aspects of solution
Produced for	Solution Development Team	Building a solution within the framework described
Approved by	Business Visionary	Responsible for delivering the required business change
	Project Manager	Responsible for ensuring the products of the project are delivered

8.2.5 Development Approach Definition

Description

The Development Approach Definition is an evolutionary product. It provides a high-level definition of the tools, techniques, customs, practices and standards that will be applied to the evolutionary development of the solution. Importantly it describes how quality of the solution will be assured. A strategy for testing and review is therefore a key part of the development approach and described in the Development Approach Definition

Roles and responsibilities

	Role	Rationale
Produced by	Technical Coordinator	Responsible for defining technical standards and ensuring development best practices are applied
Produced for	Solution Development Team	Responsible for building the solution in a professional way and to the required level of technical quality
Approved by	Project Manager	Responsible for ensuring the products of the project are delivered

8.2.6 Delivery Plan

Description

The delivery plan is an evolutionary product. It provides a high-level schedule of Project Increments and, at least for the first/imminent Increment, Timeboxes that make up that Increment. It rarely deals with task-level detail unless there are tasks being carried out by people who are not part of the Solution Development Team or before the Solution Development Team is formed.

Roles and responsibilities

	Role	Rationale
Produced by	Project Manager	Responsible for ensuring increments of the solution are delivered, predictably within agreed budget and time constraints
Produced for	All project participants and stakeholders	Everybody needs to understand at a high level what is happening when and who is participating
Approved by	Business Visionary Technical Coordinator	Responsible for ensuring that the incremental delivery of business value is optimal for the business as a whole

8.2.7 Management Approach Definition

Description

The Management Approach Definition is an evolutionary product. It reflects the approach to the management of the project as a whole and considers, from a management perspective, how the project will be organised and planned, how stakeholders will be engaged in the project and how progress will be demonstrated and, if necessary, reported. The product is outlined in Feasibility and baselined at the end of Foundations and will only evolve beyond that when circumstances change or if review of the approach identifies areas for improvement.

Roles and responsibilities

	Role	Rationale
Produced by	Project Manager	Responsible for ensuring the project is properly set up for the predictable delivery of project products
Produced for	All project participants and Stakeholders	Everybody needs to understand at a high level how the project will be managed
Approved by	Business Sponsor	Needs to be confident that the project is set up right to deliver what is needed at the right time for the right price

8.2.8 Feasibility Assessment

Description

The Feasibility Assessment is a milestone product. It provides a snapshot of the evolving business, solution and management products described above as they exist at the end of the Feasibility phase. Each of the products should be mature enough to make a sensible contribution to the decision as to whether the project is likely to be feasible or not. The Feasibility Assessment may be expressed as a baselined collection of the products or as an executive summary covering the key aspects of each of them.

Roles and responsibilities

	Role	Rationale
Produced by	Project Manager	Responsible for project management and control
Produced for	Project Governance Authority	Need to decide whether or not the project should proceed and as proposed
Approved by	Business Sponsor	The champion of the project, responsible for the return on investment

8.2.9 Foundation Summary

Description

The Foundation Summary is a milestone product. It provides a snapshot of the evolving business, solution and management products described above as they exist at the end of the Foundations phase. Each of the products should be mature enough to make a sensible contribution to the decision as to whether the project is likely to deliver the required return on investment. The Foundation Summary may be expressed as a baselined collection of the products described above or as an executive summary covering the key aspects of each of them.

Roles and responsibilities

	Role	Rationale
Produced by	Project Manager	Responsible for project management and control
Produced for	Project Governance Authority	Need to decide whether or not the project should proceed and as proposed
Approved by	Business Sponsor	The champion of the project, responsible for the return on investment

8.2.10 Evolving Solution

Description

The Evolving Solution is an evolutionary product. It is made up of all appropriate components of the final solution together with any intermediate deliverables necessary to explore the detail of requirements and the solution under construction. At any given time, such components may be either complete, a baseline of a partial solution (a Solution Increment), or a work in progress. They include, where valuable: models, prototypes, supporting materials and testing and review artefacts.

At the end of each Project Increment the Solution Increment is deployed into live use and becomes the Deployed Solution.

Roles and Responsibilities

	Role	Rationale
Produced by	Solution Development Team	Responsible for creating a solution that satisfies the requirements in the PRL
Produced for	Business Sponsor	Responsible for the return on investment
	Solution Participants	Users of the end-products of the project and part of the wider solution in live business use
Approved by	Business Visionary	Responsible for ensuring the solution that is delivered is fit for business purpose
	Technical Coordinator	Responsible for ensuring the solution that is delivered is technically fit for purpose

8.2.11 Timebox Plan

Description

The Timebox Plan is an evolutionary product that provides depth and detail for each Timebox identified in the Delivery Plan. It elaborates on the objectives provided for that Timebox and details the deliverables of that Timebox, along with the activities to produce those deliverables and the resources to do the work. The Timebox Plan is created by the Solution Development Team and is often represented on a Team Board as work to do, in progress, and done. It is updated at least on a daily basis at the Daily Stand-ups.

Roles and responsibilities

	Role	Rationale
Produced by	Solution Development Team	Responsible for self-organising to 'say what they will do'
Produced for	Solution Development Team	Responsible for self-organising and doing what they said they would do
Approved by	Project Manager Technical Coordinator	Jointly responsible for acknowledging that the team are properly focussed on the timely delivery of a fit-for-purpose Solution Increment

8.2.12 Timebox Review Record

Description

The Timebox Review Record is an evolutionary product, capturing the feedback from each review that takes place during a Timebox. It describes what has been achieved up to that point together with any feedback that may influence plans moving forwards. Where appropriate, e.g. in a regulated environment, a formal auditable record of review comments from expert Business Advisors and other roles make this a governance product.

Roles and responsibilities

	Role	Rationale
Produced by	Team Leader	Responsible for ensuring that the Iterative Development process is properly focused and controlled and that all testing and review activity is properly carried out
Produced for	Project Governance Authority	Requires assurance that development is properly controlled and that all testing and review activity is properly carried out
	Project Manager	Formally tracks progress towards delivery of the final solution
Approved by	Business Visionary	Acknowledges that the Evolving Solution Increment continues to be fit for business purpose
	Technical Coordinator	Acknowledges that the Evolving Solution Increment continues to be fit for technical purpose

8.2.13 Project Review Report

Description

The Project Review Report is a milestone product. It is typically a single document that is updated, incrementally, at the end of each Project Increment by the addition of new sections pertinent to that Increment.

At the end of each Project Increment, the purpose of this product is:

- To capture the feedback from the review of the delivered solution and to confirm what has been delivered and what has not
- To capture learning points from the retrospective for the Increment focussed on the process, practices employed and contributing roles and responsibilities
- Where appropriate to describe the business benefits that should now accrue through the proper use of the solution delivered by the project up to this point

After the final Project Increment, as part of project closure, a retrospective covering the whole project is carried out that is partially informed by the records for each Increment.

Roles and responsibilities

	Role	Rationale
Produced by	Project Manager	Responsible overall for the project and the delivery of its products
Produced for	All project participants and stakeholders and those responsible for supporting future projects (e.g.PMO)	Interested in knowing what has been achieved, the value of what has been delivered and any learning for the future.
Approved by	Business Visionary	Responsible throughout the project for ensuring the solution is fit for business purpose
	Technical Coordinator	Responsible throughout the project for ensuring the solution is technically fit for purpose
	Team Leader	Responsible throughout the project for ensuring that the Iterative Development process is properly focused and controlled and that all testing and review activity is properly carried out

8.2.14 Benefits Assessment

Description

The Benefits Assessment is a milestone product. It describes how the benefits have actually accrued, following a period of use in live operation. For projects where benefits in the Business Case are expected to accrue over a prolonged period, it is possible that a number of Benefits Assessments may be produced on a periodic basis aligned with the timeframe used for justifying the investment.

Roles and responsibilities

	Role	Rationale
Produced by	Business Visionary	Responsible for translation of the business vision into working practice
	Business Analyst	Responsible, for ensuring that benefits are assessed against Business Case and business need
Produced for	Project Governance Authority	Need to understand whether the investment in the project was justified and understand differences between predicted and accrued value
Approved by	Business Sponsor	Responsible for the return on investment

8.3 Summary

The products above are guidelines to the information needed to promote good communication within a project. They are not mandatory, and may not always be presented as documents. However, in circumstances where strong governance and/or proof of compliance with standards is important, there is real benefit to creating formal documents rather than just gaining a shared understanding (which is the normal default for DSDM). Although it may not be obvious, it is important to remember that documentation created as part of the development process and/or tied to the proactive way the project is managed, is likely to provide the most effective and robust audit trail if one is needed.

It is also critically important to remember that DSDM products are only created if and when they add value to the project and/or to the solution it creates. The most important thing is that the stakeholders and participants in the project understand what is needed and what is being delivered and that quality is assured. If documents genuinely help achieve this then create them, if not, don't waste valuable time and effort doing so.

9. Workshops

9.1 Introduction

Since organisations and information have become more complex, the Facilitated Workshop practice has been used extensively, particularly to achieve greater buy-in to decisions quickly. Organisations often achieve success (or not) through the behaviours and interactions of their people. Understanding or influencing by exerting hierarchical power is less effective and less common now than by consultation and direct relationships. As a result, enabling people to interact more effectively in a group pays enormous dividends. For many years, facilitation and Facilitated Workshops have proved to be an efficient and effective way of quickly achieving greater buy-in to decisions, solving problems, generating ideas, and action-planning.

Facilitated Workshops are a specialised type of meeting, with:

- Clear objective deliverables,
- A set of people (Participants) specifically chosen and empowered to deliver the required outcome
- An independent person (Workshop Facilitator) to enable the effective achievement of the objective

In Facilitated Workshops a neutral Workshop Facilitator guides the group through a process which enables them to work together to achieve an agreed goal; whether that be solving a problem, building a plan, gathering requirements or making decisions. The Facilitator has no stake in the outcome of the workshop and no opinion on the content. They are focused on the group dynamics and enabling the group members to collaborate to achieve their goal(s).

Facilitated Workshops ensure a team-based approach through visual and verbal communication and collaboration, where results can be achieved with speed, commitment and buy-in to the outcome.

Enabling people to communicate and collaborate effectively pays enormous dividends. Facilitated Workshops are an extremely efficient and effective way of achieving this enhanced communication. It is increasingly important for organisations to achieve success through enabling teamwork, interaction and shared understanding.

Facilitated Workshops are a proven practice: they have been used successfully throughout the business world and within DSDM for many years. As one of DSDM's core practices, they offer a way of making high quality, team-based decisions in compressed timescales. They can be used throughout the DSDM lifecycle, wherever embracing several viewpoints at the same time is advantageous, for example, when capturing and prioritising requirements, creating plans and strategies, modelling cross-functional business processes or reviewing a project deliverable or a document.

Facilitated Workshops are also a useful catalyst for effecting and supporting cultural change in an organisation, as they promote buy-in, necessitate empowerment of the Participants and require individuals to take responsibility for, and honour their commitments.

9.2 Workshop Benefits

Using facilitated workshops brings both direct and indirect benefits to a project.

- **Rapid, high quality decision-making** – Facilitated Workshops can reduce the elapsed time required to achieve objectives, such as the identification, agreement and sign-off of requirements. As all relevant stakeholders are present at the same time and able to communicate and collaborate effectively with each other, they will have greater confidence in the result. The group is focused on the objectives to be achieved in the session so that the information-gathering and review cycle is performed with greater speed. Also, misunderstandings and disagreements are made visible and can be worked out at the time, in a safe environment managed by the Workshop Facilitator. Any concerns should therefore have been raised and resolved, or noted for action after the Workshop, with appropriate people assigned as owners of specifically recorded actions.
- **Greater buy-in from all stakeholders** – Facilitated Workshops lead to Participants feeling more involved and committed to the end results due to having an opportunity to participate in, and contribute to, both the content and the decisions that are made. This builds and helps maintain enthusiasm throughout the project.
- **Building team spirit** - As well as delivering results, Facilitated Workshops are a managed way of building rapport across the community. The output of the Workshop benefits from the Participants building on each other's ideas and gaining a better understanding of each other's viewpoints. A successful Workshop depends on high levels of synergy being achieved and it is a major part of the Workshop Facilitator role to ensure an environment where this can happen.
- **Building consensus** - The Facilitated Workshop provides an opportunity for Participants to discuss the relevant subject matter, including the major issues and problems and, where possible, reach a consensus (and not compromise) on important decisions. If business procedures and practices are reviewed, Participants can gain a greater understanding of the inputs and implications of their work. This can lead to improved efficiencies, led by the Participants themselves, giving greater buy-in and commitment and therefore a greater chance of successful implementation.
- **Clarification of issues** - Workshops help to minimise ambiguities and misunderstandings. In a facilitated environment, Participants can explore and model ideas, which in turn will simplify and accelerate the review and sign-off of the Workshop deliverables.

9.3 Managing the Workshop

9.3.1 The process

This is *how* the group of people will achieve the objective. It is the responsibility of the Workshop Facilitator to plan, design and amend the process, in conjunction with the Workshop Owner, to assist the group in achieving its objective. The Workshop Owner is usually the person who asked the Workshop Facilitator to run the Workshop and who has a strong interest in its outcome.

There are a great many tools and techniques that may be used in Workshops. Physically gathering, modelling and presenting information requires the use of tools to help Participants see this information. A whiteboard, flipchart, brown paper and sticky notes are commonly used tools. Workshop techniques are used to achieve the objectives and include brainstorming, storyboards, rich pictures, SWOT analysis, grouping and many diagramming approaches.

9.3.2 Group dynamics

Group dynamics is a term used for describing how people interact together, their relationships and feelings displayed by their behaviour. This is the organic part of any group interaction. Systems and procedures do not take account of human beings with their fears, hopes, aspirations and feelings. The Workshop Facilitator's role is to manage the people through the process towards achieving the goal. Typically, this will mean making sure that individuals or factions do not dominate; ensuring shy people with valuable input are heard; ensuring discussion around issues is productive and does not become emotive or personal and keeping the group focused on the aims of the Workshop. It is the responsibility of the Workshop Facilitator to try to create the appropriate dynamic for differing situations, such as problem-solving, creativity, conflict resolution or strategic thinking and to identify and manage (and encourage the group to manage) the dynamic operating within the group. Other matters that can affect the dynamic are internal politics, pay and conditions, room layout, length of meeting, refreshments, seating and lighting. Some are within the Workshop Facilitator's control, some are capable of influence, but all need to be facilitated. An important part of the Workshop Facilitator's role is to consider all these aspects during preparation.

9.3.3 Roles in Facilitated Workshops

This section gives some guidance on which DSDM roles would fill the roles of a Workshop. Facilitated Workshop roles are defined as: Workshop Owner, Workshop Facilitator, Participants and Observer.

9.3.3.1 Workshop Owner

This person owns the objective that the Workshop is aiming to achieve and usually also owns the budget to run the Workshop. It is up to the Workshop Owner to set the objectives and deliverables of the Workshop, although the Workshop Facilitator should help the owner in clarifying and scoping these. The Workshop objectives should also be understood and agreed by the Participants at the start of the Workshop.

The owner of a project kick-off Workshop may be the Business Sponsor, whereas the owner of a Timebox planning Workshop could be the Team Leader or even the Business Ambassador. What is important is that the Workshop Owner is involved in the definition and resourcing of the Workshop and retains ownership of the objective throughout.

9.3.3.2 Workshop Facilitator

The Workshop Facilitator manages the process and dynamic of the Workshop, enabling the Participants to concentrate on the content and the deliverables. The Workshop Facilitator should be neutral to the Workshop objectives, the deliverables (outcome) of the Workshop and the Participants. He/she is responsible for helping the group to meet the Workshop objectives. Ideally, the Workshop Facilitator should come from outside the project to ensure - and signify - neutrality. Some organisations have internal facilitators that are allocated to Workshops and other organisations employ external consultants. If the Workshop Facilitator is from within the project, it is important that their behaviour is also seen by the group as remaining independent of the outcome.

The Workshop Facilitator's skills and abilities include:

- Listening effectively and accurately
- Summarising
- Observing and recalling conversation and behaviour
- Communicating clearly
- Identifying similarities and differences between statements
- Recognising and understanding different viewpoints and perspectives
- Assessing content and information for relevance
- Identifying assumptions
- Recognising effective and ineffective behaviour
- Intervening appropriately as necessary
- Being a model of effective behaviour
- Providing feedback impartially and tactfully
- Accepting feedback calmly
- Being in control of own behaviour and using own behaviour effectively
- Developing trust with and within groups

The Workshop Facilitator may engage a Scribe to record and publish the Workshop outputs. To support or speed up the process there may be more than one person allocated to this role in a Workshop, for example, sometimes a Technical Scribe is used where models and documentation are to be created directly into a specific toolset. Usually the Scribe is not a Participant, since it is difficult to participate fully at the same time as scribing outcomes and decisions.

9.3.3.3 Participant

A Participant is chosen because they are needed to produce the deliverables or achieve the objectives of the Workshop. Participants must add value to the Workshop. To do this they need to have the knowledge, skills and experience to be able to contribute to the objective of the Workshop and be empowered to make decisions. Group facilitation is a lean process so only the people essential to achieving the objectives and deliverables should be there. In order to make the group dynamics as effective as possible extra/uninvited Participants should be avoided, as larger groups exponentially increase the number of possible communication channels.

A Participant could perform one of many roles within the organisation or may represent suppliers or customers from outside. They may hold any of the DSDM roles, including Advisor roles.

9.3.3.4 Observer

Observer is an optional role, with no direct input to the production of the Workshop's immediate deliverables. The Observer gains from attending and hearing the discussions, but is silent and has no influence on or input into these discussions. Typically the Observer sits outside the Participant group so that they do not distract the active Participants. The Workshop Facilitator should pay attention to the effect on the general group dynamic when Observers are present and manage any impact they may have.

Examples of the use of the Observer role could include:

- Auditors of the Workshop process or the Workshop Facilitator's competence
- A trainee Workshop Facilitator who wants to observe the group dynamics without being part of the group
- An auditor of the project processes
- Solution Developers and Solution Testers gaining understanding and background around requirements being defined by the business roles
- Support staff gathering useful background information

Whenever an Observer is present in a Workshop, they should not contribute towards the content, process or deliverables. If they need to take an active part, they should be invited and acknowledged as Participants.

9.3.4 Facilitated Workshop activities

The key activities within a Facilitated Workshop are:

- Define and plan the Workshop
- Prepare for the Workshop
- Facilitate the Workshop session
  - Run the Workshop
  - Workshop retrospective
- Document the outcome in a Workshop Report if required
- Follow-up with post-Workshop actions and review

9.3.4.1 Define and plan the Workshop

The Workshop Owner, with support from the Workshop Facilitator, defines the objectives of the Workshop, nominates the Participants and agrees, in outline, the form that the Workshop should take. It may sometimes be necessary to define several Workshops to achieve the objectives. Workshops can be effective with any number of Participants, from 4 to 100+. A Workshop needs a minimum of 4 Participants to create effective group dynamics. Significant planning and structure will be required for larger Workshops, which may include the use of co-facilitators and possibly splitting the Participants into groups.

9.3.4.2 Prepare for the Workshop

In preparation for the Workshop, the Workshop Facilitator circulates information to the Participants in advance so that they fully understand the objective of the Workshop and the background to it, and have time to prepare. An agenda detailing when and where the Workshop will take place, who will be attending and the order of proceedings, will be sent out, together with any pre-Workshop reading. In particular, individuals will be advised where their input to the Workshop is needed so that they may prepare the information that they need to make an effective contribution, and where necessary, collect the views of those they are representing.

9.3.4.3 Facilitate the Workshop - run the Workshop

The tight timescales of a DSDM project mean that the Workshop needs to maintain its focus and pace.
One ground rule some Workshop Facilitators operate is the principle of the five-minute rule wherein any disagreement that cannot be resolved in a period of five further minutes is parked as an 'open issue'. Such open issues are documented and deferred to a later session or possibly taken outside the Workshop for resolution.

For Workshops to be effective, there are a few basic guidelines which the Workshop Facilitator should define and agree with the group and which can be highlighted to bring people back on track, should it become necessary. Sample guidelines (ground rules) are:

- Please be on time - as timescales are constrained
- Respect the views of others
- One conversation at a time
- Each individual in the group has a responsibility to maintain focus
- Phones/technology off/silent

9.3.4.4 Facilitate the Workshop - Workshop retrospective

The effectiveness of the Workshop should be examined before the end of the session and any lessons learned fed back into the operation of future Workshops. In particular, did the Workshop meet its objectives fully and did all Participants contribute to the process? Most importantly, how effective did the Participants feel that the Workshop had been e.g. did it run to time?

9.3.4.5 Document the Workshop

The Workshop Scribe should produce and distribute the output very soon (usually within 48 hours) after the Workshop, to all Participants and, if appropriate, to other interested parties who will be affected by the output of the Workshop. This report should be brief and should document:

- Decisions
- Actions with action owners
- Open issues
- The output of the Workshop itself, as appropriate
- And sometimes the process used

It does not record minutes or verbatim statements

9.3.4.6 Workshop follow-up

The satisfaction of the Workshop Owner with the Workshop's results should be confirmed. All actions marked for follow-up activity outside the Workshop forum must be addressed, not just documented! The responsibility for taking the actions lies with the Participants and the Workshop Owner.

9.4 Success Factors for Facilitated Workshops

The factors which have been found, in practice, to greatly improve the success of Facilitated Workshop are:

- An effective, trained, independent Workshop Facilitator
- Flexibility in the format of different Workshops, but clearly defined objectives
- Thorough preparation before the Workshop, by Workshop Facilitator and Participants
- A mechanism for ensuring that the outcomes of previous Workshops are built in, where appropriate
- Decisions and agreements that are not forced. If the Participants cannot agree on a point within the Workshop (perhaps due to lack of information or time), the Workshop Facilitator should recognise this outcome and elicit from the group the appropriate action to remedy the shortfall
- Participants receiving a Workshop report, describing decisions, actions and the outcome of the Workshop, very soon after it has been run

Much can be learned by scheduling a short retrospective just before the end of each Workshop and documenting the benefits and concerns from the Workshop. Sometimes this is also useful at key points during long Workshops e.g. before lunch.

9.5 Other Types of Workshop

Sometimes project Workshops are run without a Workshop Facilitator, although such Workshops can be difficult to manage. However, if Participants are familiar with Facilitated Workshops they can be reminded of the process and encouraged to act as they would if it were facilitated. If Participants are unfamiliar with Facilitated Workshops, as a minimum it would be useful to review the ground rules suggested in this session for applicability. It would also be useful at the beginning of the Workshop for the leader to suggest a few minutes discussing how it should be run.

9.6 Summary

Facilitated Workshops are one of DSDM's key practices. The skill, independence and neutrality of the Workshop Facilitator are important to ensure successful workshops. DSDM describes the Workshop roles, together with the Workshop activities (before, during and after the workshop). DSDM also describes the benefits of using Facilitated Workshops.

If you would like more information about facilitation a good place to start is: **www.iaf-world.org/**

10. MoSCoW Prioritisation

10.1 Introduction

In a DSDM project where time has been fixed, it is vital to understand the relative importance of the work to be done in order to make progress and keep to deadlines. Prioritisation can be applied to requirements/User Stories, tasks, products, use cases, acceptance criteria and tests, although it is most commonly applied to requirements/ User Stories. (User Stories are a very effective way of defining requirements in an Agile style; see later chapter on Requirements and User Stories for more information.)

MoSCoW is a prioritisation technique for helping to understand and manage priorities. The letters stand for:

- **M**ust Have
- **S**hould Have
- **C**ould Have
- **W**on't Have *this time*

The use of MoSCoW works particularly well on projects. It also overcomes the problems associated with simpler prioritisation approaches which are based on relative priorities:

- The use of a simple high, medium or low classification is weaker because definitions of these priorities are missing or need to be defined. Nor does this categorization provide the business with a clear promise of what to expect. A categorisation with a single middle option, such as medium, also allows for indecision
- The use of a simple sequential 1,2,3,4… priority is weaker because it deals less effectively with items of similar importance. There may be prolonged and heated discussions over whether an item should be one place higher or lower

The specific use of Must Have, Should Have, Could Have or Won't Have *this time* provides a clear indication of that item and the expectations for its completion.

10.2 The MoSCoW Rules

10.2.1 Must Have

These provide the Minimum Usable SubseT (MUST) of requirements which the project guarantees to deliver. These may be defined using some of the following:

- No point in delivering on target date without this; if it were not delivered, there would be no point deploying the solution on the intended date
- Not legal without it
- Unsafe without it
- Cannot deliver a viable solution without it

Ask the question 'what happens if this requirement is not met?' If the answer is 'cancel the project – there is no point in implementing a solution that does not meet this requirement', then it is a Must Have requirement. If there is some way around it, even if it is a manual and painful workaround, then it is a Should Have or a Could Have requirement. Categorising a requirement as a Should Have or Could Have does not mean it won't be delivered; simply that delivery is not guaranteed.

10.2.2 Should Have

Should Have requirements are defined as:

- Important but not vital
- May be painful to leave out, but the solution is still viable
- May need some kind of workaround, e.g. management of expectations, some inefficiency, an existing solution, paperwork etc. The workaround may be just a temporary one

One way of differentiating a Should Have requirement from a Could Have is by reviewing the degree of pain caused by the requirement not being met, measured in terms of business value or numbers of people affected.

10.2.3 Could Have

Could Have requirements are defined as:

- Wanted or desirable but less important
- Less impact if left out (compared with a Should Have)

These are the requirements that provide the main pool of contingency, since they would only be delivered in their entirety in a best case scenario. When a problem occurs and the deadline is at risk, one or more of the Could haves provide the first choice of what is to be dropped from this timeframe.

10.2.4 Won't Have *this time*

These are requirements which the project team has agreed will not be delivered (as part of this timeframe). They are recorded in the Prioritised Requirements List where they help clarify the scope of the project. This avoids them being informally reintroduced at a later date. This also helps to manage expectations that some requirements will simply not make it into the Deployed Solution, at least not this time around. Won't Haves can be very powerful in keeping the focus at this point in time on the more important Could Haves, Should Haves and particularly the Must Haves.

10.3 MoSCoW Relating to a Specific Timeframe

In a traditional project, all requirements are treated as Must Have, since the expectation is set from the start that everything will be delivered and that typically time (the end date) will slip if problems are encountered. DSDM projects have a very different approach; fixing time, cost and quality and negotiating features. By the end of Foundations, the end dates for the project and for the first Project Increment are confirmed.

In order to meet this commitment to the deadline, DSDM projects need to create contingency within the prioritised requirements. Therefore the primary focus initially is to create MoSCoW priorities for the project. However, when deciding what to deliver as part of the Project Increment, the next focus will be to agree MoSCoW priorities for that Increment. So at this point, a requirement may have two priorities; MoSCoW for the project and MoSCoW for the Increment. Finally, when planning a specific Timebox (at the start of each Timebox) the Solution Development Team will allocate a specific priority for the requirements for this Timebox. At this point, the majority of requirements are Won't Have (for this Timebox). Only requirements that the Solution Development Team plan to work on in the development timebox are allocated a Must Have, Should Have or Could Have priority.

Therefore requirements may have three levels of priority:

- MoSCoW for the project
- MoSCoW for the Project Increment
- MoSCoW for this Timebox

> *For example:*
>
> *even if a Must Have requirement for an IT solution is the facility to archive old data, it is very likely that the solution could be used effectively for a few months without this facility being in place. In this case, it is sensible to make the archive facility a Should Have or a Could Have for the first Project Increment even though delivery of this facility is a Must Have before the end of the project.*
>
> *Similarly, a Must Have requirement for a Project Increment may be included as a Should Have or a Could Have (or a Won't Have) for an early Timebox.*

It is important that the bigger picture objectives (completion of the Project Increment and delivery of the project) are not forgotten when working at the Timebox level.

One simple way to deal with this is to create a separate Timebox PRL, a subset of the project PRL that is specifically associated with an individual Timebox and leave the priorities unchanged on the main PRL for the project.

10.4 Ensuring effective prioritisation

10.4.1 Balancing the priorities

When deciding the effort allocated for Must Have requirements, remember that anything other than a Must Have is, to some degree, contingency, since the Must Haves define the Minimum Usable SubseT which is guaranteed to be delivered.

DSDM recommends:

- Getting the percentage of project/Project Increment Must Haves (in terms of effort to deliver) to a level where the team's confidence to deliver them is high – typically no more than 60% Must Have effort
- Agreeing a pool of Could Haves for the project/Project Increment that reflects a sensible level of contingency - typically around 20% Could Have effort. Creating a sensible pool of Could Haves sets the correct expectations for the business from the start – that these requirements/User Stories may be delivered in their entirety in a best case scenario, but the primary project/Project Increment focus will always be on protecting the Must Haves and Should Haves

This spread of priorities provides enough contingency to ensure confidence in a successful project outcome.

NB When calculating effort for a timeframe, Won't Haves (for this timeframe) are excluded.

DSDM's recommendations reflect a typical project scenario. The important thing to make MoSCoW work is to have some visible flexibility in the level of requirements which must be delivered.

The safe percentage of Must Have requirements, in order to be confident of project success, is not to exceed 60% Must Have effort.

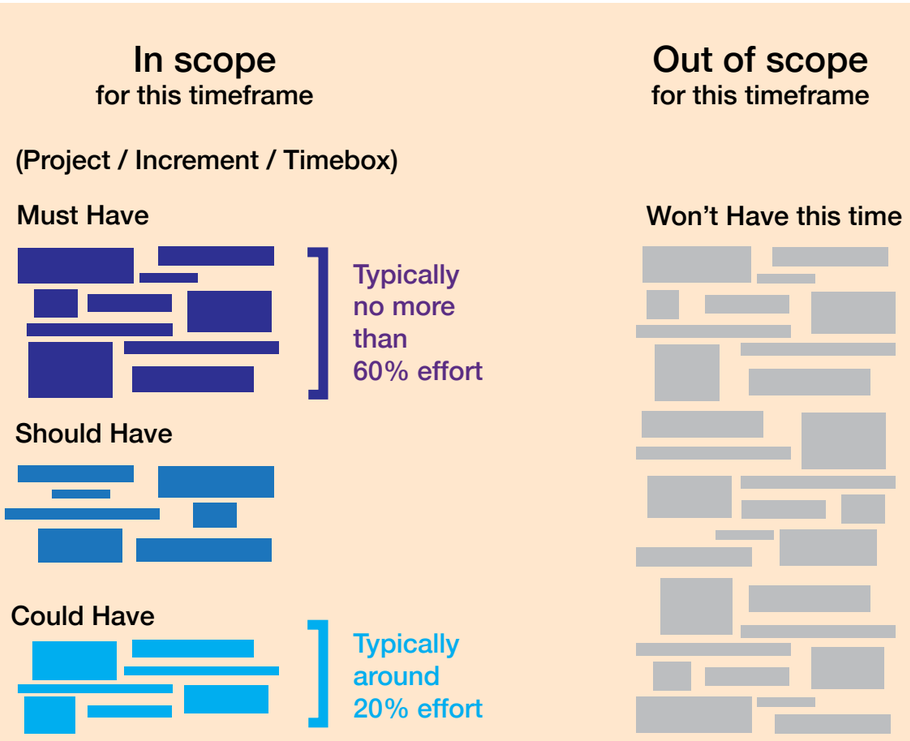


Figure 10a: MoSCoW – balancing priorities

Levels of Must Have effort above 60% introduce a risk of failure, unless the team are working in a project where all of these criteria are true:

- Estimates are known to be accurate
- The approach is very well understood
- The team are "performing" (based on the Tuckman model)
- The environment is understood and low-risk in terms of the potential for external factors to introduce delays

In some circumstances the percentage of Must Have effort may be significantly less than 60%. However this can be used to the benefit of the business, by providing the greatest possible flexibility to optimise value delivered across a larger proportion of Should Haves.

The exact split of effort between Musts, Shoulds, and Coulds is down to each project team to agree, although DSDM also recommends creating a sensible pool of Could Haves, typically around 20% of the total effort. Effective MoSCoW prioritisation is all about balancing risk and predictability for each project.

10.4.2 Agreeing up front how priorities will work

DSDM defines what the different priorities mean – the MoSCoW Rules. But whereas the definition of a Must Have is not negotiable, the difference between a Should Have and a Could Have can be quite subjective. It is very helpful if the team agree, at the start of their project, how these lower level priorities will be applied. Understanding in advance some objective criteria that separate a Should Have from a Could Have and ensuring that all roles on the project buy into what has been agreed can avoid much heated discussion later. Look for defined boundaries that decide whether a requirement is a Should Have or a Could Have?

> *For example:*
>
> *At what point does the number of people impacted raise a Could Have to a Should Have? Or, What value of benefits would justify dropping this requirement from a Should Have to a Could Have?*

Ideally this agreement is reached before the requirements are captured.

10.4.3 When to prioritise

Every item of work has a priority. Priorities are set before work commences and the majority of this prioritisation activity happens during Foundations. However, priorities should be kept under continual review as work is completed. As new work arises, either through introduction of a new requirement or through the exposure of unexpected work associated with existing requirements, the decision must be made as to how critical it is to the success of the current work using the MoSCoW rules. When introducing new requirements, care needs to be taken not to increase the percentage of Must Have requirement effort beyond the agreed project level. The priorities of uncompleted requirements should be reviewed throughout the project to ensure that they are still valid. As a minimum, they should be reviewed at the end of each Timebox and each Project Increment.

10.4.4 Discussing and reviewing priorities

Any requirement defined as a Must Have will, by definition, have a critical impact on the success of the project. The Project Manager, Business Analyst and any other member of the Solution Development Team should openly discuss requirements prioritised as Must Have where they are not obvious Must Haves ("Without this would we cancel the project/increment?"); it is up to the Business Visionary or their empowered Business Ambassador to explain why a requirement is a Must Have.

The escalation of decision-making processes should be agreed early on, e.g. Business Ambassador and Business Analyst to Business Visionary to Business Sponsor, and the level of empowerment agreed around decision-making at each level.

At the end of a Project Increment, all requirements that have not been met are re-prioritised in the light of the needs of the next Increment. This means that, for instance, a Could Have that is not met in one Increment may be reclassified subsequently as a Won't Have for the next Increment, because it does not contribute enough towards the business needs to justify its inclusion. However, it could just as easily become a Must Have for the next Increment, if its low priority in the first Increment was based on the fact it was simply not needed in the first Solution Increment.

10.5 Using MoSCoW to Manage Business Expectations

The MoSCoW rules have been defined in a way that allows the delivery of the Minimum Usable SubseT of requirements to be *guaranteed*. Both the Solution Development Team and those to whom they are delivering share this confidence because the high percentage effort of Shoulds and Coulds provides optimum contingency to ensure delivery of the Must Haves.

The business roles can certainly expect more than delivery of only the Must Haves. The Must Haves are guaranteed but it is perfectly reasonable for the business to expect delivery of more than the Minimum Usable SubseT in the timeframe, except under the most challenging of circumstances.

DSDM's recommendation to create a sensible pool of Could Have contingency – typically around 20% of the total project/increment effort - identifies requirements that are less important or which have less impact if not delivered, in order to protect the more important requirements. This approach implies that the business can reasonably expect the Should Have requirements to be met, in addition to all of the Must Haves. It also implies that in a best case scenario, the Could Have requirements would also be delivered.

The Solution Development Team cannot have the confidence to guarantee delivery of all the Must Have, Should Have and Could Have requirements, even though these have all been estimated and are included in the plan. This is because the plan is based on early estimates and on requirements which have not yet been analysed in low-level detail. Applying pressure to a team to guarantee delivery of Musts, Shoulds and Coulds is counter-productive. It usually results in padded estimates which give a false perception of success. "We always achieve 100% (because we added significant contingency to our figures").

So, combining sensible prioritisation with timeboxing leads to predictability of delivery and therefore greater confidence. This also protects the quality of the solution being delivered. Keeping project metrics to show the percentage of Should Haves and Could Haves delivered on each Project Increment or Timebox will either re-enforce this confidence, if things are going well, or provide an early warning of problems, highlighting that some important (but not critical) requirements may not be met at the project level.

10.6 How does MoSCoW Relate to the Business Vision

10.6.1 The Business Sponsor's perspective

The starting point for all projects is the business vision. Associated with the business vision are a set of prioritised requirements that contribute to delivery of the vision. Also associated with the business vision is a Business Case that describes the project in terms of what value it will deliver back to the business. Depending on the organization, this Business Case may be an informal understanding or it may be defined formally, showing what Return On Investment (ROI) is expected in order to justify the cost of the project.

The MoSCoW priorities are necessary to understand the Minimum Usable SubseT and the importance of individual requirements. The Business Visionary must ensure that the requirements are prioritised, evaluated in business terms, and delivered to provide the ROI required by the Business Case, in line with the business vision.

10.7 Making MoSCoW Work

Requirements are identified at various levels of detail, from a high-level strategic viewpoint (typically during Feasibility) through to a more detailed, implementable level (typically during Evolutionary Development). High-level requirements can usually be decomposed to yield a mix of sub-requirements, which can then be prioritised individually. This ensures the flexibility is maintained, so that if necessary, some of the detailed less important functionality can be dropped from the delivered solution to protect the project deadline.

It is this decomposition that can help resolve one of the problems that often confront teams: that all requirements appear to be Must Haves.

If all requirements were genuinely Must Haves, then the flexibility derived from the MoSCoW prioritisation would no longer work. There would be no lower priority requirements to be dropped from the deliverables to keep a project on time and budget. This goes against the DSDM ethos of fixing time and cost and flexing features (the triangles diagram in the Philosophy and Fundamentals chapter). Believing everything is a Must Have is often symptomatic of insufficient decomposition of requirements.

Remember that team members may cause scope creep by working on "interesting" things rather than the important things. MoSCoW can help avoid this.

10.8 Tips for Assigning Priorities

1. Ensure that the business roles, in particular the Business Visionary and the Business Analyst, are fully up to speed as to why and how DSDM prioritises requirements.
2. Consider starting with all requirements as Won't Haves, and then justify why they need to be given a higher priority.
3. For each requirement that is proposed as a Must Have, ask: 'what happens if this requirement is not met?' If the answer is 'cancel the project; there is no point in implementing a solution that does not meet this requirement', then it really is a Must Have. If not, then decide whether it is Should Have or a Could Have (or even a Won't Have this time).
4. Ask: 'if I come to you the night before Deployment and tell you there is a problem with a Must Have requirement and that we can't deliver it – will you stop the Deployment?' If the answer is 'yes' then this is a Must Have requirement. If not, decide whether it is Should Have or a Could Have.
5. Is there a workaround, even if it is a manual one? If a workaround exists, then it is not a Must Have requirement. When determining whether this is a Should Have or a Could Have requirement, compare the cost of the workaround with the cost of delivering the requirement, including the cost of any associated delays and any additional cost to implement it later, rather than now.
6. Ask why the requirement is needed – for this project and this Project Increment.
7. Is this requirement dependent on any others being fulfilled? A Must Have cannot depend on the delivery of anything other than a Must Have because of the risk of a Should Have or Could Have not being delivered.
8. Allow different priorities for acceptance criteria of a requirement.

   For example:

   'The current back-up procedures need to ensure that the service can be restored as quickly as possible.' How quick is that? Given enough time and money, that could be within seconds. A smarter definition would be to say it Should happen within four hours, but it Must happen within 24 hours.

9. Can this requirement be decomposed? Is it necessary to deliver each of these elements to fulfil the requirement? Are the decomposed elements of the same priority as each other?
10. Tie the requirement to a project objective. If the objective is not a Must Have, then probably neither is the requirement relating to it.
11. Does the priority change with time? For example, for an initial release a requirement is a Should Have, but it will become a Must Have for a later release.
12. Prioritise testing, using MoSCoW.
13. Use MoSCoW to prioritise your To Do list. It can be used for activities as well as requirements.

10.9 Summary

MoSCoW (Must Have, Should Have, Could Have, Won't Have *this time*) is primarily used to prioritise requirements, although the practice is also useful in many other areas. On a typical project, DSDM recommends no more than 60% effort for Must Have requirements on a project, and a sensible pool of Could Haves, usually around 20% effort. Anything higher than 60% Must Have effort poses a risk to the success and predictability of the project, unless the environment and any technology is well understood, the team is well established and the external risks minimal.

11. Iterative Development

11.1 Iterative Development Overview

Iterative development is a process in which the Evolving Solution, or a part of it, evolves from a high-level concept to something with acknowledged business value.

Each cycle of the process is intended to bring the part of the solution being worked on closer to completion and is always a collaborative process, typically involving two or more members of the Solution Development Team.

Each cycle should:

- Be as short as possible, typically taking a day or two, with several cycles happening within a Timebox
- Be only as formal as it needs to be - in most cases limited to an informal cycle of *Thought, Action and Conversation*
- Involve the appropriate members of the Solution Development Team relevant to the work being done. At its simplest, this could be, for example, a Solution Developer and a Business Ambassador working together, or it could need involvement from the whole Solution Development Team including several Business Advisors

Each cycle begins and ends with a *conversation* (in accordance with DSDM's Principles *collaborate* and *communicate continuously and clearly*). The initial *conversation* is focussed on the detail of what needs to be done. The cycle continues with *thought* - a consideration of how the need will be addressed. At its most formal, this may be a collaborative planning event, but in most cases *thought* will be limited to a period of reflection and very informal planning. *Action* then refines the Evolving Solution or feature of it. Where appropriate, *action* will be collaborative. Once work is completed to the extent that it can sensibly be reviewed, the cycle concludes with a return to *conversation* to decide whether what has been produced is good enough or whether another cycle is needed. Dependent on the organisation and the nature of the work being undertaken, this *conversation* could range from an informal agreement to a formally documented demonstration, or a "show and tell" review with a wider group of stakeholders.

As Iterative Development proceeds, it is important to keep the agreed acceptance criteria for the solution, or the feature of it, in clear focus in order to ensure that the required quality is achieved without the solution becoming over-engineered. An agreed timescale for a cycle of evolution may also help maintain focus, promote collaboration and reduce risk of wasted effort.

11.2 Planning Iterative Development

During Foundations, it is very important to decide on a strategy for development that encompasses how the potentially large problem of evolving a solution can be broken down into manageable chunks for delivery in Timeboxes. DSDM identifies two ways in which this may be considered. The first describes a requirement focus, the second describes a solution focus. It is important to note that neither is better than the other and equally valid alternative ways of approaching Iterative Development may be available; the key is to find the most appropriate development strategy for the project through discussion. DSDM does not dictate how the strategy should be formed and agreed, just that there should be a strategy for development.

11.2.1 Requirement focus

DSDM identifies that the requirements can be of three different types:

- Functional
- Usability
- Non-functional

(Note that traditionally, usability is considered to be a type of non-functional requirement but is elevated into a class of its own because of its importance to the business user of the final solution and to facilitate business interaction with the rest of the Solution Development Team in this important aspect of solution design)

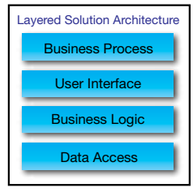
An individual cycle of Iterative Development or even the work of a Timebox may focus on evolving the solution to meet one or more of these requirement types.

On a simple feature, a cycle may encompass all three perspectives at the same time. However, where iterative development of a feature involves many cycles, involving several different people, the team may decide to focus a cycle on one or perhaps two specific perspectives rather than covering all of them at the same time.

For example, the team may decide to focus early cycles on the functional perspective of a requirement – ensuring, and demonstrating, that the detail of the requirement is properly understood and agreed. This may be followed by cycles focussed on usability – ensuring interaction with the solution is effective and efficient. Later cycles may then focus on ensuring the required non-functional quality (e.g. performance or robustness) is achieved, that all tests pass as expected, and all acceptance criteria for the feature are met.

11.2.2 Solution focus

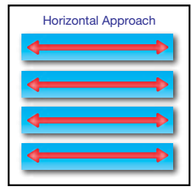
DSDM identifies that the solution may be considered to have a number of architectural layers. The example used here relates to a business system development but the concept can also be applied to a non-IT project, such as a marketing campaign.



Iterative Development may follow an approach in which Timeboxes deliver horizontal slices of the solution, vertical slices or a combination of the two.

Horizontal Approach

The horizontal approach considers the solution layer by layer with each Timebox incrementally delivering increased complexity of business process or layers of complexity/completeness as regards technology.



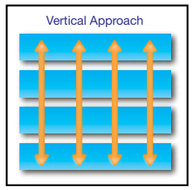
The advantage of the horizontal approach is that it allows an initial sight of the full breadth of the solution very early on. The disadvantage is that nothing works fully until the last horizontal slice is delivered. Therefore no business benefit can accrue until that point.

For example:

Projects where a horizontal approach may be appropriate:

- *Project H1 is delivering a corporate intranet. At the end of the first Timebox, the business want to see a thin but complete first cut of the solution that demonstrates the full breadth of what will be delivered. Subsequent Timeboxes deliver additional layers of more detailed information*
- *Project H2 is using a mixture of technologies. It is agreed to develop the user interface first and then to add business logic and finally to hook the solution into a custom-designed database*

Vertical Approach



The vertical approach slices through multiple layers of the solution with each Timebox delivering one or more fully functional features.

The advantage of the vertical approach is that delivery of prioritised features may enable Solution Increments to be more quickly and frequently deployed into live use. The disadvantage is that the full breadth of the solution is not clear until late in the project.

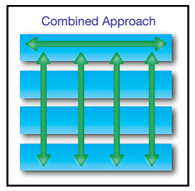
For example:

Projects where a vertical approach may be appropriate:

- *Project V1 is delivering a simple ecommerce solution and wants to have a fully working group of requirements available at the end of each Timebox ready for deployment. For example, a first timebox making items available for purchase visible to the user, a second Timebox allowing items to be selected and put in a basket etc*
- *Project V2 is enhancing an existing system by adding new features with each new feature being fully built and tested within the boundaries of a single Timebox and then immediately deployed into live use.*

Combined Approach

The combined approach starts by focussing one or two Timeboxes on (thin) horizontal slices of the solution in order to fully understand the breadth of the solution before reverting to a vertical approach to incrementally deliver fully functional features.



The advantages of the combined approach are that there is an initial view of the overall solution early in the project, and incremental delivery of business value into live use is also achieved. There are no obvious disadvantages.

For example:

Projects where a combined approach may be appropriate:

- *Project C1 is a business change project in the operations department of a company that is planning to simultaneously implement new business processes supported by new technology. Early Timeboxes focus on the design of the end-to-end business process that, whilst subject to change at the detail level, will act as an anchor for the development of the supporting technology. Subsequent Timeboxes deliver the technology to support the process and the overall solution (new process and new technology) is deployed into live use.*
- *Project C2 is automating an existing business process. The first Timebox prototypes simple user screens to show how the new system will support the end-to-end process. Having got this accepted, subsequent timeboxes can deliver working components against the context of the agreed framework.*

11.3 Controlling Iterative Development

Each Iterative Development cycle is intended to result in an incremental change to the Evolving Solution that brings it, or more probably a feature of it, progressively closer to being complete or *"done"*.

It is quite possible, however, that the review of such an incremental change may reveal that the solution has evolved in a way that is not right. Under such circumstances it is important, wherever possible, to have the option to revert to the previously agreed version and undertake another cycle based on what has been learnt and on any new insight arising from the review. Configuration Management of the solution is therefore an important consideration.

11.4 Quality in Iterative Development

One of the defining principles of DSDM is to *never compromise quality.* To achieve this we should (amongst other things) define a level of quality in the early lifecycle phases and then ensure that this is achieved. The challenge then is how to meaningfully define quality and then measure it in an iterative context.

11.4.1 Quality criteria

Quality criteria need to deal with required characteristics of the product/feature with these being driven by the context in which the product/feature is going to be used.

For example:

For a house brick

Its physical dimensions and ability to bear load are clearly important characteristics

- *If the dimensions are inconsistent (too long or too short) then a uniform wall cannot be built from a random sample of bricks*
- *If the load-bearing capacity is not sufficient then the building will be unsafe*
- *If the load-bearing capacity exceeds the need then it is likely that the bricks will cost more to make*

In terms of context, bricks for use in colder climates need to be thicker than those used in temperate ones.

For example:

For a software application

A key characteristic may be the ability to carry out business processes within specified time constraints (e.g. a 3-second response time) and perform calculations to a high level of accuracy (e.g. +/- 1% accuracy).

In terms of context, web sites selling tickets suffer spikes of extreme load when a major event goes on sale and need to be extra resilient because of this.

11.4.2 Acceptance criteria

By the time Iterative Development of the solution starts, the main deliverables will already have some acceptance criteria associated with them from the Foundations phase. Although it might not be practical or even appropriate to get to this level of detail during Foundations, by the time development of a particular feature starts acceptance criteria should be objective and quantified (rather than subjective).

For example:

An 'on-line purchase' feature would need a defined set of inputs (e.g. product codes and purchase volumes), a planned response (e.g. calculating and displaying the item and total costs whilst separately calculating tax) and the context within which this is happening (e.g. checking the stock levels needed to fulfil the request).

If the acceptance criteria are vague or subjective (as may be the case at the end of Foundations) then more *conversation* is needed to agree on the specific details.

Note that this information informs both what needs to be built and how it will be assessed, so it is essential it is done before work starts.

Thought is applied to both how the solution is built and how to verify that it has been built correctly. Where the DSDM structured timebox is used (Chapter 13.3), the detailed work on acceptance criteria takes place primarily in the Investigation step. Where a less formal structure is used, it takes place as a first step in addressing a particular requirement once it has been selected, whenever that occurs within the Timebox.

11.4.3 Validation and verification

Validation asks 'Are we building the right thing?' whilst verification asks 'Are we building the thing right?'

In an Iterative Development context, validation does not need to be a separate activity as the process of collaborative design of the solution with direct involvement of Business Ambassador and Business Advisor roles means this happens naturally. However, verification activity still needs to be explicitly considered, to ensure it is fully integrated in the Iterative Development cycle. How this will be achieved is part of the development approach and should be described in the Development Approach Definition if appropriate.

Having agreed how the quality will be verified *action* ensures verification is carried out effectively and efficiently. The person responsible for producing the product will naturally carry out his or her own assessments as part of that development activity. Simultaneously, a separate person (in most cases the Solution Tester) needs to prepare for the independent verification activity. This can be just as time-consuming as making the product (in some cases more time-consuming).

11.4.4 Static and dynamic verification (reviews and testing)

There are two broad classes of verification – static and dynamic. Static verification involves inspecting a deliverable against its acceptance criteria or agreed good practice. The advantage of this type of verification is that it is based on observation alone and so could not cause harm to the product being inspected.

Some items can only be statically verified.

> *For example:*
>
> *Documents, or the dimension of a house brick*

Other items might be verified statically and/or dynamically.

> *For example:*
>
> *The components of an engine can be reviewed against the blue print to statically test them. Running the engine allows it to be dynamically tested.*

As static methods present no risk that the deliverable will be damaged by the process (presuming non-invasive methods), there are potential advantages to inspection even when an item could be tested dynamically.

Static verification - reviews

Reviews can range from informal peer reviews through to highly structured and formal reviews involving experts or perhaps groups of people. The level of formality is often driven by the nature of the product and by corporate or regulatory standards.

Some types of static testing can be automated.

> *For example:*
>
> *On an IT project, code scanners can check that agreed coding standards have been followed. This can be particularly useful for initially checking security aspects of a solution.*

The Technical Coordinator is responsible for the technical quality of the solution by "ensuring adherence to appropriate standards of technical best practice" and so should ensure that:

- Such practices are appropriate and understood by all
- An appropriate regime of peer review, with an appropriate level of formality, is part of the teams working practice
- Appropriate, contemporary evidence of review activity is captured as required

Whilst the rigour of a review can vary, mostly they share certain key qualities:

1. There are key roles that need to be fulfilled:
   - The producer(s) of the item being reviewed (author(s) in the case of a document)
   - The reviewers
   - A review moderator, where appropriate, for very formal reviews
2. All reviews require time to be carried out. A simple, informal review may be peers gathering at a desk and reviewing the item together, whereas a formal review needs to properly planned, preferably as a Facilitated Workshop
3. Every review involves assessing the product against criteria, which may be specific to that item (defined as acceptance criteria) or general to an item of that kind (general standards or good practice such as those defined by the development approach). The criteria need to be agreed (and probably documented) in advance to gain the most benefit, but they can of course also evolve over time to accommodate the current situation and appropriate innovation in working practice
4. Every review must reach a conclusion. Commonly there are three potential results:
   - The item is fit for purpose and no further action is required
   - The item needs minor amendments to make it fit for purpose. In this case, the review group might nominate one individual to check that required changes are made
   - The item needs major amendments before it is fit for purpose. In this case the item typically needs to be fully reviewed again after being reworked

Reviews may occasionally result in no clear outcome. In this case the people involved need to collaborate and if necessary bring other people in to the discussion in order to reach an agreed outcome.

Dynamic verification - testing

The act of dynamically checking an item is commonly known as testing. There are three broad classes of tests which are useful to consider when dynamically verifying a deliverable:

- Positive tests check that a deliverable does what it should do
  - e.g. when you add an item to your basket on a web site the item does appear there
- Negative tests check that a deliverable doesn't do what it shouldn't do
  - e.g. if you put the wrong key into a padlock, you shouldn't be able to open it
- 'Unhappy path' tests check the behaviour of the deliverable when unusual or undefined things happen
  - e.g. what happens when a car engine overheats? And is that behaviour ok?

Based on these test classes, with some thought it is usually possible to think up multiple positive, negative and unhappy path tests for each acceptance criterion. Every test would typically have the following:

- A defined starting state
- A defined set of actions which we will carry out
- An outcome which we expect to see

Planned and exploratory testing

Preparing defined tests in advance has useful benefits:

- Firstly, you can ensure that you consider each of the acceptance criteria and testing perspectives to give you the best chance of identifying high-value tests
- Secondly, you can prioritise your tests (using MoSCoW) to make sure that you cover the most important ones during execution
- Finally, you can prioritise the test execution, for example to target areas of risk

Combining these factors (good coverage, highest value criteria and highest areas of risk) ensures that the best project value is derived from the testing activity

It is possible to test without preparation (this is typically known as exploratory testing) but it is a technique that requires considerable skill and experience with testing and extensive procedural knowledge of the way the solution will be used in order to be effective. It is critically important that any defective behaviour identified as

a result of exploratory testing can be replicated. This means that the starting state, the actions taken and the results (both actual and expected) still need to be defined but this is often difficult to achieve because it has to happen *after* the problem has been identified. For this reason, a balance between planned and exploratory testing is advisable. See below for further considerations on the use of exploratory testing.

Manual and automated testing

In recent years, test automation tools have improved significantly. These now range from shareware, through integrated development environments, to wide-ranging commercial tools. Given the rapid pace associated with Iterative Development, the effective and efficient use of automation is essential for testing.

Automated tools are very effective when dealing with precise inputs and predictable outputs, but are less effective when such precision cannot be achieved and/or when judgement is required about whether a deviation from what is expected is acceptable or not. For these reasons, a blend of manual and automated testing is usually required.

Automated tests can take considerable time to prepare (though the best tools minimise this) and are used to run the same tests time and again. However, it is important not to become blindly reliant on automated tests, because by running the same tests the same types of problems appear, whilst other errors outside these tests remain undetected. Consequently, there is still value in manual testing to quickly test new or specific aspects of the solution. Tests should only be automated if there is a high degree of confidence that they will need to be repeated multiple times (though in Iterative Development this is highly likely in most cases). Manual testing should focus more on exploratory aspects (exploratory testing) whilst automated testing is typically a form of planned testing. When exploratory tests find a significant problem they should be reverse engineered into a planned test and then automated where practicable.

Documentary evidence

Whatever methods of testing are used, in some circumstances it may be essential or mandatory to capture evidence to show what was done and what behaviour was observed. Review of test results may reveal trends that can be addressed through evolving either the solution design or development standards. Very importantly, it also allows a project to demonstrate due diligence to an auditor or external regulator, if appropriate.

Roles and responsibilities for testing activity

The Technical Coordinator for the project is responsible for the overall quality of the solution from a technical perspective and so is responsible for "ensuring that the non-functional requirements are reasonable at the outset and subsequently met".

The Solution Tester is responsible for carrying out all types of testing except for:

- Business acceptance testing: that is the responsibility of Business Ambassador(s) and Business Advisors
- Unit testing of the feature: that is the responsibility of the Solution Developer

Note: As part of a collaborative team, the Solution Tester will be supporting other roles to fulfil their testing responsibilities by providing testing knowledge and expertise.

11.4.5 Are we 'done' yet?

After verification has been undertaken (whether statically or dynamically) the key question is whether or not the acceptance criteria have been met in a meaningful way. It may be obvious that the criteria have clearly been met or have clearly failed. In other cases more *conversation* is needed to decide whether the team are confident that the solution is fit for purpose or not, based on what has been observed. This could imply that the acceptance criteria were not sufficiently understood or defined. Alternatively, even though all the 'Must' criteria may have been met, the product may have failed against so many other, lesser criteria that it is unlikely it will actually deliver the benefit needed from it.

If the solution has unconditionally met all the acceptance criteria, then it is 'done'. Where only some of the criteria have been met, then the product may need to be evolved further to ensure that more criteria are met next time it is validated. More *conversation* will be needed, followed in due course by more *thought* and *action* to implement what has been agreed upon.

Once an item is agreed as 'done' it is advisable to record any discretionary acceptance criteria which remained unfulfilled. This information is valuable as:

- It may have a knock-on effect on other parts of the solution or work yet to be done
- The team may choose to spend time later on fixing these criteria rather than implementing lower-value features

Remember that an accumulation of less serious defects may eventually have an impact on the Business Case which is not clearly shown by any one failed criterion in isolation.

11.5 Summary

Iterative Development in a project context needs up-front thought. However this is not about big design up-front (BDUF) or detailed planning. It is more a consideration of the strategy for development. The DSDM philosophy states that projects *deliver frequently* so the 'big picture' needs to be broken down into small manageable chunks to facilitate this frequent delivery.

The principles: *focus on the business need, deliver on time, collaborate* and *never compromise quality* must also be considered as these drive how the Solution Development Team works. How work is planned at the detailed level, ensuring the right people are involved at the right time and assuring quality as development progresses, requires the whole team to be bought in to a sensible strategy for development that they help shape during the Foundations phase. Where appropriate, this will be documented in the Development Approach Definition.

For small, simple projects delivering conceptually simple solutions, consideration of these issues may take an hour or two and be based on simple conversation and 'round the table' agreement. However, as a general rule, the need for a more formal and more carefully considered Iterative Development strategy increases with the size and complexity of the project organisation and the solution to be created.

Quality assurance is a key part of delivering a solution that is fit for business purpose. However, the formality and rigour of testing will depend very much on the nature of the project and the standards laid down by the organisation.

12. Modelling

12.1 Introduction

One of the main sources of errors and failures in projects is cited as the lack of good communication and the consequential misunderstandings between different stakeholder groups. Each stakeholder group (customers, users, managers, developers, technical experts) typically has its own particular jargon, which can lead to confusion and misinterpretation.

Modelling techniques are designed to improve communications and prompt the right questions. They provide an early means of checking that the solution being developed is what is required. They are a valuable aid to achieving project success. The purpose of modelling requirements is:

- To improve understanding through visual representations
- To support transparency by simplifying core elements of a requirement, usually in a picture
- To abstract the most relevant information for clarity (see below)
- To allow cross-checking for consistency

Many organisations benefit from the use of models, prototypes and mock-ups to establish requirements, to confirm expectations and to test the achievability of objectives. These models can be as different as:

- A storyboard to represent a television advertisement
- Architectural blue-prints to define a housing development
- An artist's impression of a landscaped park
- A scale model of a car to be built
- Process diagrams to establish the required functionality to be supported by a software solution
- A network diagram showing components of a communications network

For example:

In a business change project:

- *Process maps may be drawn to establish existing processes and their inter-relationships (the as-is process)*
- *A further set of process maps may be drawn to define how the new processes will work (the to-be process)*
- *The to-be model may be validated by walk-throughs or role play*
- *Later, IT functionality to support the business process may be visually represented in a simple way in order to check understanding of the requirement, for example as one or more screens (although it will probably not yet be working in the way the final solution will work)*
- *These visual representations will then be iteratively refined and agreed with representatives of those who will use them*
- *The visual representations become fully working in the final solution, with logic being built behind the screens to make them function*
- *The to-be model is used to drive User Guides on how to work using the new business process*

12.2 What is a Model?

A model can be defined as:

- A description or analogy used to help visualise something that cannot be directly observed
- A small but exact copy of something
- A pattern or figure of something to be made

Models may be physical (a built version of some part of an eventual solution - a prototype) For example, a prototype of working software or may be expressed in a specialised language. For example, a diagramming convention, with its own rules and symbols.

DSDM has the principle *communicate continuously and clearly,* and advocates using visual communication. This means choosing the appropriate medium for communication, with a focus on the right type of communication for the target audience, used at the right time in the project, and using a format and style that can be understood.

Modelling helps to make elements of the solution visible as early as possible. Examples of this could be the use of diagrams, or acting out a new process to be supported by a new IT system. However, the amount of time and effort put into a model should only be just enough to satisfy the purpose of the model and no more.

To enhance clarity, modelling usually incorporates some degree of ***abstraction***, which involves omitting certain information from the model to allow clearer focus on another specific aspect. A different diagram, for a different purpose and different target audience may be used to show these other features. It is usually far too complex and confusing to try to address several target audiences in a single model.

For example:

The map of the underground railway/subway in a city shows just what it needs to, in order to communicate specific information to its target audience (the traveller). The underground/subway map allows travellers to move from Station A to Station B, and therefore only shows the stations and the links between them. It omits everything else, including the most rudimentary aspects of geography that would indicate how close together stations on different lines are..

If the underground railway/subway is closed and the traveller is sent above ground (for example in an emergency) a different model will be needed (a road map, showing as a minimum compass direction and distances).

Modelling, like communication in any form, is used to answer specific questions. As a minimum the project always needs to know:

- Who is this for?
- What do they need it for?

12.3 Modelling and Prototyping

Modelling and prototyping are linked concepts. A prototype is always a kind of model; a model is not necessarily a prototype. For example, we can model an existing situation: a building; a staffing structure; a database structure.

A *prototype* usually implies a new structure.

In IT, the term *model* has traditionally been used to refer to a set of diagrams.

12.3.1 Target audience for the model

It is important that the level of detail and the language used in the model is appropriate for its target audience. In DSDM, models are used to communicate between teams of mixed specialisms (business, technical, etc.). It is vital to consider the effectiveness of any particular modelling approach to the whole target audience.

12.3.2 The perspectives for modelling

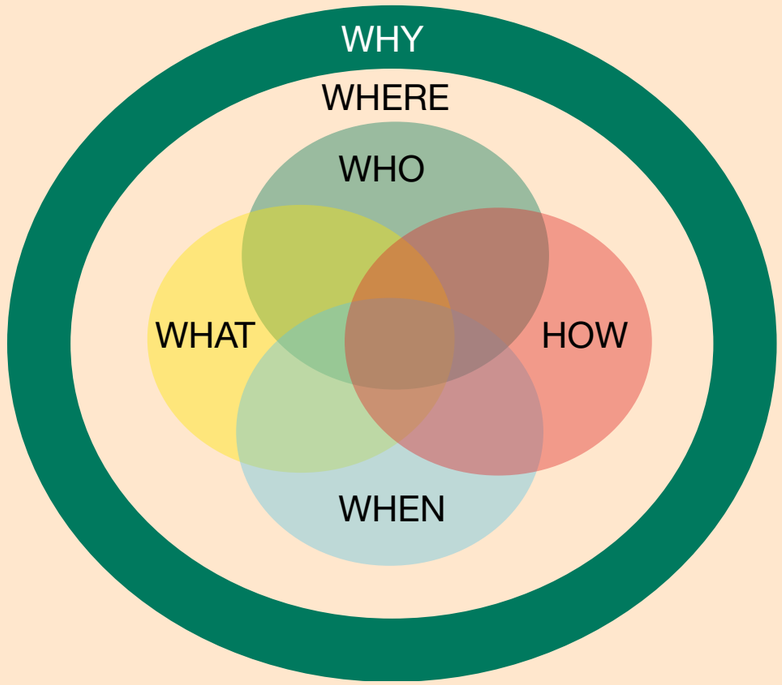


Figure 12a: Modelling perspectives

A coherent picture of a solution area can be gained by considering the perspectives: "what", "where", "when", "how", "who" and "why", and the relationships between them. For example: who performs which processes; what data is needed to support each process. Matrices can be helpful in drawing these relationships.

The table below shows an interpretation of these perspectives from an IT system point of view, but parallels can easily be drawn for other types of project.

WHAT	The information within the solution area, data, relationships and business rules.
HOW	The functions, features and processes within the solution area
WHERE	The locations at which the business operates, in relation to the solution area
WHO	The people: customers, users, stakeholders
WHEN	The events of importance to the business (times and scheduling)
WHY	The business objectives and strategy, as related to the project

DSDM does not require the drawing of models to cover every perspective, unless there is value in doing so.

However, it is worth checking during a project whether any perspective has been missed (overlooked) rather than intentionally omitted.

During the Foundations phase of the project, when a high-level Prioritised Requirements List is being compiled, models can help the whole team be clear about the scope and the boundaries of the solution. One modelling example for foundations is the use of user story mapping.

A user story map is a visual way of making a link between the User Stories (see Chapter 17 - Requirements and User Stories), to allow focus on the bigger picture and to see how the individual User Stories relate to one another. This visualisation then allows the stories to be organised to reflect priorities and their relationships to one another, to form the basis for creation of the Delivery Plan and Timebox Plans.

There are many modelling techniques to choose from, some focusing on a business viewpoint, others used purely for IT. Some of the commonly used techniques are currently:

- Storyboards
- Flowcharts
- Swim-lane diagrams
- Process models
- Class models (IT)
- Use case diagrams (IT)

12.4 Modelling in the Agile Lifecycle

The level of modelling at each phase of the DSDM lifecycle must be appropriate to the level of complexity and characteristics of the project/programme in question.

12.4.1 Pre-project

Pre-Project, existing high-level models may be useful to illustrate how this project, or the solution it could deliver, would fit into a wider picture of change, for example as part of a larger programme.

12.4.2 Feasibility

During the Feasibility phase, models are likely to support a relatively simple 'big picture' view of what is being proposed and are used to explore possibilities and help communicate options. Feasibility prototypes may be used to help establish what is possible from a technical perspective as well as helping visualise what a solution might look like from a business perspective.

12.4.3 Foundations

During the Foundations phase, more precise and elaborate models may be created. These models will help communicate plans and intentions to a variety of audiences. Models of business process and business organisation (as they are today and as they are proposed in the future) may be valuable as might high-level designs of technical solutions such as system architecture models and data models. At this point models and prototypes can be used to help clarify scope and support high-level planning by revealing omissions, inconsistencies and dependencies.

12.4.4 Evolutionary Development

During the Evolutionary Development phase it is likely that, where they add ongoing value, high-level models will continue to be evolved in terms of depth and detail as a way of helping to explore the detail of requirements and ways that these may be met as part of the Evolving Solution. Where appropriate, models may be developed that will help with Deployment and with the ongoing operation and support of the solution in live use.

12.4.5 Deployment

It is unlikely that new models will be created during this phase but some that were created to help with the Deployment will be used at this time and perhaps refined for future Project Increments where applicable. In addition, models created to help operate and support the soluton may be refined as it transitions into live use.

12.4.6 Post-Project

In post-project, the models used help to operate and support the solution will continue to be refined in line with any changes to the Deployed Solutuon over time.

12.4.7 Progressive business change

As the solution is deployed, the "as is" models of the current situation give way to the "to be" models which represent the new product or service. It is necessary to make a clear distinction between these two similar models and to be explicit about which is being modelled to avoid later confusion.

It is usually advisable to keep "as is" models as simple as possible unless there is a strong reason to do otherwise: for example, if the detail is needed to support business change initiatives, or to understand the transition work. The Business Visionary and Business Ambassador roles are embodiments of such "as is" information and should be available throughout the project and during the transition to the new ways of working. The Business Analyst will have the modelling skills to evolve such useful diagrams. The Business Visionary and Business Ambassadors' availability helps to limit the amount of excessive detail which can sometime obscure such models and reduce their effectiveness.

12.5 Conclusion

Whatever the product or business solution being developed, DSDM recommends an iterative, incremental and collaborative approach to modelling, following the DSDM lifecycle. This approach relies heavily on effective communication.

- DSDM advocates clear and continuous communication, using rich communication techniques. Modelling is one of DSDM's key practices to support effective communication. Models should be developed iteratively, taking a top-down approach through to the detail and modelling from different perspectives
  - Models should always be an aid and never a bureaucratic overhead. The aim of models is to enhance the effectiveness of communication for all members and levels of the development process
  - The choice of model should be influenced by the intended audience; use models they will understand
- The use of models, and the formality with which they are created and reviewed, depends on the reason for the model, the nature of the project and on the skills and experience of the team
  - The level of models required to support the building of a new power station will be significantly different from the level of models required to support development of a small web-site
- DSDM does not advocate any particular modelling techniques, although there are some well-defined standard modeling approaches. The simplest rules are:
  - Do what works for the project and the organisation; capitalise on the skills which exist within the organisation
  - Use diagrams and models to establish a common language between the teams
  - Do enough appropriate modelling and no more
- Modelling is intended to help people visualise complex things
  - Models can help clarify the overall picture at a high level
  - Models can help break down the project into comprehensible blocks of work
  - Models can be used as a basis for increment and timebox planning

The overriding goal for DSDM is development of a working solution, or a partial solution, as soon as possible. However, an appropriate amount of design up front (EDUF) supported by a few well-chosen models and prototypes at the appropriate time can save the cost of expensive communication errors.

13. Timeboxing

13.1 Introduction

Timeboxing is one of DSDM's key practices.

DSDM defines a Timebox as a fixed period of time, at the end of which an objective has been met. The Timebox objective is usually completion of one or more deliverables. This ensures the focus for a Timebox is on achieving something complete and meaningful, rather than simply "being busy". At the end of a Timebox, progress and success is measured by completion of products (requirements or other deliverables) rather than completing a series of tasks.

The optimum length for a Timebox is typically between two and four weeks – long enough to achieve something useful, short enough to keep the Solution Development Team focused. On a very short and rapidly moving project, it is possible to have a Timebox as short as a day. In exceptional circumstances, a Timebox might be as long as six weeks. The disadvantage of longer Timeboxes is that the team may lose focus. By evolving the solution through a series of short Timeboxes, the team is able to more frequently assess their true progress – "What have we delivered?" If their progress is not meeting their own expectations, this provides early warning of problems, and an early opportunity to address the problems.

Timeboxing is more than just setting short time periods and partitioning the development work. It is a well-defined process to support the creation of low-level products in an iterative but controlled fashion. Timeboxing incorporates frequent review points to ensure the quality of those products and the efficiency of the Iterative Development process.

By managing on-time/on-target delivery at the lowest level, on-time and on-target delivery at the higher levels can be assured. Initial MoSCoW prioritisation of the work for the Timebox and continual re-assessment of what can be achieved in its agreed timeframe ensures that timeboxes finish on time every time and deliver a working solution to meet business objectives in line with business expectations - "No nasty surprises".

The Project Increment and the entire project can also be considered as Timeboxes, as they share the characteristics of delivering a fit-for-purpose solution in a pre-set timeframe. These higher-level Timeboxes are managed through the control applied at the lower level – the Timebox. Unless qualified by Project or Increment, the word Timebox will always refer to a Timebox during the Evolutionary Development of the phase.

13.2 Timebox Options

Every timebox begins with a kick-off and ends with a close-out. Beyond this, DSDM recognises two styles of timebox:

- A DSDM structured Timebox
- A free format Timebox

The choice of Timebox style may be driven by factors such as the availability of the Business Ambassador and other business roles or the type of product being developed.

13.3 A DSDM Structured Timebox

This is the original DSDM-style Timebox, which provides a standard, repeatable internal structure to a Timebox.

The structure within a Timebox is very useful to allow forward planning of the times when the Business Ambassador will attend specific planning, feedback and review sessions. As well as these specific planned sessions, there is still an expectation of some day-to-day business engagement, e.g. attending Daily Stand-ups and timely response to urgent questions. By projecting this structure forward to future Timeboxes, it becomes possible to schedule the various Timebox control points (kick-off, the three reviews, close-out) for all the Timeboxes in the Project Increment. Where a Business Ambassador is trying to manage a very busy diary, this can be a great help.

The DSDM structured Timebox also provides a framework plan for the timebox, focussing Iterative Development activity within the Timebox to ensure convergence on an accurate business solution. With this structure, the Solution Development Team know when they should have completed their initial investigations (by the end of week one in a three week Timebox), they know when they should be nearing a conclusion and have their product nearly ready (by the end of week two in a three week Timebox). With this structure, the Solution Development Team know that the final few days are focused on final tweaks and fine-tuning, to ensure the Timebox closes cleanly. At any point during the DSDM structured Timebox, the whole Solution Development Team has visibility of progress and early warning if the overall Timebox objectives are at risk.

A DSDM structured Timebox comprises three main steps:

- Investigation
- Refinement
- Consolidation

Each of these steps ends with a review

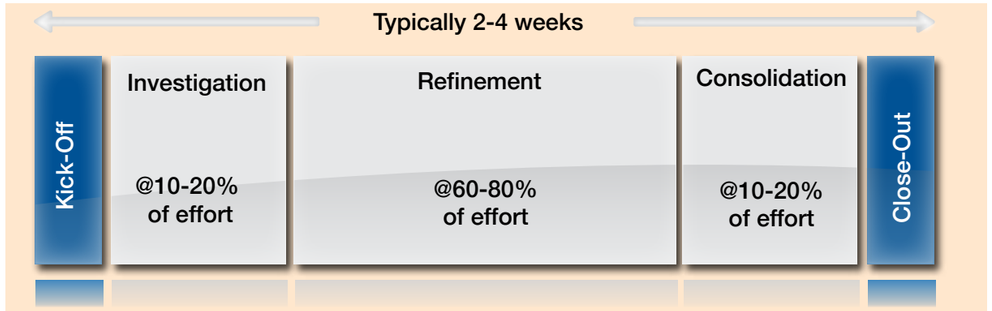


Figure 13a: A DSDM structured timebox

Timebox	Nature of the work done	Suggested timescale
Kick-off	Short session for the Solution Development Team to understand the timebox objectives and accept them as realistic	Approx 1-3 hours for a 2-3 week Timebox
Investigation	The Investigation includes confirmation of the detail of all the requirements and all the products to be delivered by this timebox. Includes agreement on: • the Timebox deliverables • the acceptance criteria for the deliverables • the measures of success for the Timebox Investigation ends with a review which informs Refinement and may be a valuable governance touch-point	Approx. 10-20% of Timebox
Refinement	Encompasses the bulk of the development, addressing requirements and testing (technical and business) the Timebox products, in line with agreed priorities Refinement ends with a review which informs Consolidation and may be a valuable governance touch-point	Approx. 60-80% of Timebox
Consolidation	Ties up any loose ends related to Evolutionary Development and ensures all products meet their previously agreed acceptance criteria Consolidation ends with a review, which informs Close-out and may be a valuable governance touch-point	Approx. 10-20% of Timebox
Close-Out	Formal acceptance of the Timebox deliverables by the Business Visionary and Technical Coordinator. This is followed by a short Timebox retrospective workshop, to learn from the Timebox and to take actions to improve future Timeboxes	Approx 1-3 hours for a 2-3 week Timebox

13.3.1 Timebox Kick-off

The aim of the Timebox Kick-off is to:

- Review the Timebox objectives, as outlined in the Delivery Plan, to gain a common understanding of what is to be achieved
- Ensure that it is still feasible within the Timebox to deliver what was initially expected during the Foundation phase, and to re-plan accordingly if this is no longer possible
- Where possible, agree the acceptance criteria for each product to be delivered within the Timebox.
  - If it is not possible to agree this level of detail at the Timebox Kick-off, then agreement can be deferred to the end of Investigation but, in this case, high-level acceptance criteria must be agreed until the additional detail is available. (Going into a Timebox without an understanding of the acceptance criteria is extremely risky)
- Review the availability of all members of the Solution Development Team (including the business roles) to participate in Timebox activities for this Timebox
  - Commitment to delivery is based on pre-agreed resource levels at the project level. However an individual's availability can vary between one Timebox and another, for example, due to planned time off
- Highlight any known dependencies (internal or external) that may affect this Timebox. The Solution Development Teams dependencies could be
  - Internal - other Solution Development Team's on this project working concurrently in parallel Timeboxes
  - External - people or projects outside the team's control that may impact this project

The Kick-off should be attended by all members of the Solution Development Team (including Business Ambassador(s)) who will be working in the Timebox as well as the Project Manager, the Technical Coordinator and the Business Visionary.

13.3.2 Step 1: Investigation

The aim of Investigation is to provide a firm foundation for the work to be carried out during Refinement and to clarify further the requirements and their acceptance criteria. Investigation entails the Solution Development Team jointly investigating the detail of requirements and agreeing how these requirements will be met as part of the Evolving Solution. This detailed information may be captured as part of acceptance criteria, against individual requirements, or as an elaboration of the Prioritised Requirements List.

Acceptance criteria should be confirmed as correct and providing appropriate coverage of the scope of each requirement. Whenever possible, an initial model or prototype of the solution is created to demonstrate an understanding of the requirements and to provide early visibility of the solution for assessment and feedback.

During Investigation, the entire team should work together on the full set of requirements agreed for the Timebox at the kick-off. It is necessary to understand the detail and priorities of the work intended for completion in the Timebox, so that informed decisions can be made later about which lower priority requirements may be dropped if necessary.

Some very early testing may be possible and is to be encouraged, but during investigation the main test focus is to work with the Business Ambassador and the Business Analyst, as well as the rest of the Solution Development Team, to clarify acceptance criteria and to start planning testing for this Timebox.

At the end of investigation the whole Solution Development Team review the following:

- Dependencies:
  - The team ensure they understand any dependencies within this Timebox on teams working in other parallel timeboxes on this project (concurrent teams) or elsewhere in the business, and between the requirements they are addressing
- Timebox Plan:
  - The team informally review the work still to be done, and agree which members of the team will be working on what. This ensures that no single individual is overloaded. This informal review validates the Timebox Plan (or highlights if the Investigation work has shown that the Timebox Plan is no longer viable, so that remedial action can be taken (see Chapter 11- Iterative Development for full detail)
- Risks:
  - Based on the information gained from the investigation, and risks recognised for this Timebox from the Delivery Plan and risk log, the Solution Development Team analyse the risks associated with this Timebox and, on that basis, ensure an acceptable balance of requirements of differing priorities in accordance with the MoSCoW rules

The feedback from this review is captured as a Timebox Review Record (which can be as simple as a brief email, confirming what has been agreed). The investigation feedback is used to drive the next step in the Timebox – Refinement – and ensures the Solution Development Team can fully commit to achieving the Timebox objectives, based on an enhanced level of understanding.

A formal, documented review involving Business or Technical Advisors with responsibility for legislative or corporate compliance may be used as a demonstrable form of control over the Evolving Solution and provide a valuable audit trail.

13.3.3 Step 2: Refinement

The aim of Refinement is to complete as much of the development work as possible, including testing the products(s). Development and testing are carried out iteratively; the primary objective is to meet the detailed acceptance criteria previously agreed (at the latest, by the end of Investigation) but also to keep the focus on the current business need. The order of the work should be driven by the MoSCoWed priorities for this Timebox but should be influenced by other factors, such as:

- A sensible development order from a technical perspective
- Availability of specific resources such as Technical Advisors, Business Advisors
- Any known cross-team dependencies

Refinement ends with a review with the Business Ambassador(s) and, where appropriate, other stakeholders, such as Business Advisors who have been actively involved in this Timebox, and the Business Visionary. By this point (end of Refinement) the work for this Timebox should be nearly ready.

The review determines what actions are necessary to achieve full completion of the work according to the acceptance criteria by the end of the Timebox. No new work should be started after this point. Final feedback (fixing minor outstanding issues) requested at this time should be carefully considered and prioritised. Any significant demand for change at this point often exposes a lack of appropriate involvement of business roles previously during this Timebox - a lesson to be learned for the future.

This review would typically involve a demonstration of the product developed within the Timebox. The feedback from this review is captured as a Timebox Review.

Again, with appropriate formality, this review can be an effective demonstration of legislative or corporate compliance.

13.3.4 Step 3: Consolidation

During Consolidation, the actions agreed at the Refinement review are carried out, together with final testing and any work required to satisfy organisational or project standards. Examples of this could be holding a peer review, or migrating code to a different environment. Any final quality control checks are carried out by the Solution Development Team to ensure all products or requirements/User Stories meet the business need to an acceptable quality. Consolidation ends with a review to check whether the Timebox objectives have been met. Any products not meeting the agreed acceptance criteria by this point (the end of the Timebox) are deemed not to have been delivered. These undelivered products remain open on the Prioritised Requirements List.

Formal sign off here, or during close-out, by qualified advisors will acknowledge compliance of the solution with corporate or legislative needs.

13.3.5 Timebox Close-out

The primary aim of the Close-out is to record formal sign-off or acceptance of all the products delivered from this timebox. An important secondary aim is to determine what is to be done about work that was initially part of the timebox but was not completed. Such work may be:

- Considered for the next Timebox
- Scheduled for some point later in the Project Increment or project
- Dropped from the Project Increment or project

If overall timescales are to be met, it is important to avoid the situation where unfinished work passes automatically into the next Timebox, without any consideration of the overall priorities.

A final aim of the Close-out is to look back on the Timebox, to see if there is anything that can be learned to make the Iterative Development process and/or Timebox management process more effective in the future. This on-going process of holding a short retrospective workshop as part of each Timebox close-out has a number of benefits:

- To allow the team to learn from their experiences in this Timebox
  - To recognise and build on the good experiences
  - To recognise problems and avoid repeating the same mistakes in the future
  - To define issues to be resolved in the next Timebox
- To gather ongoing information for use in the later, more formal reviews (at the end of the Project Increment and at the end of the project)

Where the Timebox has been successful and where the team is already established, this retrospective workshop can be very short. If there have been problems during the Timebox, or if this is the first Timebox with a new team, the retrospective workshop may need additional time.

Depending on the time needed for the Close-out, it may be practical and sensible to run the Close-out back to back with the Kick-off session for the next Timebox.

13.4 A Free Format Timebox

The free format Timebox reflects the style used by other popular Agile approaches such as a Scrum Sprint. A free format Timebox may be effective where the formality and structure of the DSDM structured Timebox is not possible or helpful.

The free format Timebox also starts with a Kick-off and finishes with a Close-out. However in between, there may be any number of formal or informal review points. Typically the Solution Development Team will pick up one or more products or requirements/User Stories and evolve these iteratively until completed. Completion means a product of requirement/User Story meets the previously agreed acceptance criteria. The Solution Development Team then pick up the next product or requirements/User Stories and repeat the process. This free format style relies on the Business Ambassador being available consistently to review and provide feedback on an on-going basis.

Figure 13b: A free format timebox

Within this chapter, the use of this graphic represents a link to other Agile approaches.

Timebox	Nature of the work done
Kick-off	Short session for the Solution Development Team to understand the timebox objectives, to agree what work (requirements and products) will be taken on in this Timebox and agree their Timebox priorities, and to accept this workload as realistic
Iterative Development	Iterative Development and testing of individual requirements/ User Stories and other products, as agreed in the Kick-off and in a sequence driven by the agreed priorities for this Timebox It may still be appropriate to informally adopt the concepts of Investigate, Refine, Consolidate to converge on the accurate solution for each individual requirement/User Story or Product, for example understanding the lower level detail and agreeing the acceptance criteria at the start of the development of a requirement/User Story. This use of Investigate, Refine, Consolidate per requirement/User Story helps to mitigate the risk of too many iterations as the product is elaborated It is still important that reviews are scheduled during the body of the free format Timebox, to maintain business focus and stakeholder buy-in
Close-Out	Formal acceptance of the Timebox deliverables by the Business Visionary and Technical Coordinator. This is followed by a short Timebox retrospective workshop, to learn from the Timebox and to take actions to improve future Timeboxes

13.5 The Daily Stand-Up

A key and integral part of all Timeboxes, regardless of the style adopted, is the Daily Stand-up. This is the Solution Development Team's opportunity to share information across the team and to do any day-to-day re-planning and reorganising necessary when issues occur. However, it is important to emphasise that ongoing informal communication goes on between all team members during the day as needed, and not just at the Daily Stand-up.

On a daily basis, the Solution Development Team get together for a Stand-up session. The Stand-up usually takes place at the same time and same place each day (with the Timebox plan visible), so that others who are not part of the Solution Development Team may listen in. Normally facilitated by the Team Leader, the Stand-up is a daily opportunity for everyone to understand progress against objectives at a detailed level and to expose issues and blockers that may be getting in the way.

The Stand-up:

- Has the following active participants :
  - All members of the Solution Development Team including Business Ambassador(s)
  - Any Business Advisors actively involved in this Timebox
  - Any Technical Advisors actively involved in this Timebox
- Typically uses a simple format in which each participant in turn describes:
  - What I have been doing since the last stand-up that helps achieve the Timebox objectives
  - What I will be doing between now and the next stand-up to help achieve the Timebox objectives
  - What problems, risks or issues (blockers) I have that will prevent me or the team achieving the Timebox objectives

- Has a short and fixed duration – normally no longer than 15 minutes
  - 2 minutes per participant + 2 minutes is a good guide
- Is ideally held with all participants standing in a circle by their Team Board; this is sometimes called an Information Radiator
- Re-enforces the desire to keep the session short and informal and to keep everyone focused
  - Teleconference Stand-ups (dial-ins) may be necessary where the team is split across multiple sites. However, for choice, this works better if groups at each site get together in a room and dial in to the groups at the other site(s). For teleconference Stand-ups, it becomes even more important to use the suggested format (as bulleted above) to provide a simple structure for the communications. For teleconference Stand-ups, the team need to decide how the Team Board will be used. In these circumstances, it may be an electronic version, rather than a physical area
- May be attended by other roles e.g.
  - The Business Visionary – in order to keep in touch with progress, to provide on-going visible support
  - The Project Manager – in order to observe progress and pick up escalated issues
  - The Technical Coordinator – in order to keep abreast of technical decisions and pick up escalated technical issues

It is important that the Stand-up is used to identify problems and to agree who needs to participate in solving any problems that arise. The Stand-up should not attempt to solve these problems if reaching a resolution will take any more than a minute or two. It is common practice for problems to be taken off-line. This allows for follow-up discussions which are not run under stand-up guidelines and which only involve those who are directly impacted.

The Stand-up also provides the primary mechanism for the Solution Development Team to track progress and exert the necessary flexibility and control over their work to ensure on-time delivery of the agreed products by the end of the Timebox.

Daily Stand-ups are also an effective technique when used outside a Timebox, for example during Foundations, or in any circumstance where informal and on-going communication needs to be embedded as part of "the way we do things".

13.6 Dealing with Change within a Timebox

Iterative Development is what enables a team to deliver a product that is genuinely fit for its intended purpose by the end of a Timebox. Converging on the accurate solution is achieved through constant refinement of the product, based on review by the business, led by the Business Ambassador and supported by the Business Analyst. It is vital that the decisions about whether, at any given time, a solution appears right, or needs to change to make it right, are both quick and sure. If decision-making is not quick and sure, there is a real risk that significant time will be lost (by waiting for decisions to be made) or wasted (as a result of decisions being overturned). It is important that all members of the Solution Development Team are appropriately empowered to handle any change that falls within the agreed scope of the Timebox objectives, without the need for a formal change control process that reaches beyond the team.

As a rule of thumb, the following scenarios always mean a change of scope and therefore need more formal management (outside the empowerment of the Solution Development Team):

- Changing the breadth of the solution (i.e. adding to the high-level requirements or removing Must Have requirements)
- Increasing the percentage of Must Have effort, either by introduction of new Must Have requirements, or by upgrading Won't Have, Could Have or Should Have priorities to Must Haves

However, negotiation around the detail (the depth) of the solution can generally be handled by the empowered Solution Development Team without the need for any escalation or formal approval by those outside the Solution Development Team

Regardless of whether changes are deemed to impact scope or not, typically the Solution Development Team is empowered to operate within agreed boundaries without the need to escalate to the Project Manager or other project-level roles.

For example:

Following the practice of MoSCoW, dropping a Could Have requirement from a Timebox (or even from an Project Increment or project) is normally something that is reported after the event, rather than requiring permission. By comparison, significantly changing the meaning of a Must Have requirement often requires external guidance.

However, all changes to the content of a Timebox must be agreed and accepted by the Solution Development Team as a whole, and must not be simply imposed on them or decided by one individual member of the Solution Development Team in isolation.

Boundaries of empowerment should be established by the end of Foundations and reviewed for effectiveness on a regular basis (as a minimum, at the end of each Timebox).

13.7 Timeboxes – The Wider Context

Application of the timeboxing practice (described above) in conjunction with the practice of MoSCoW prioritisation ensures each Timebox delivers a fit-for-purpose product in the agreed timeframe. As each Timebox delivers fit-for-purpose products on time, then this indicates that the Project Increment is on track and thus the project as a whole is on track.

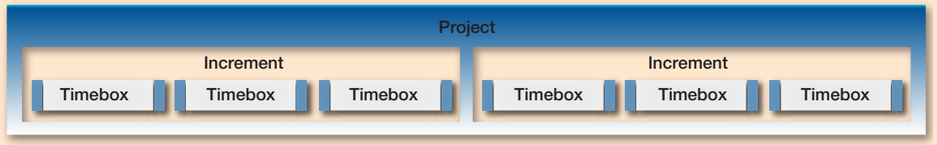


Figure 13c: DSDM timeboxes in the context of increments and project

13.8 Summary

Timeboxing is one of DSDM's key practices and is used in combination with the practice of MoSCoW prioritisation to ensure on-time delivery. At the lowest level, the Timebox maintains focus on delivery in the short term (weeks or even days). This provides control at the lowest level, as well as a clear indication of the health of the project overall. If Timeboxes are successfully delivering the Must Haves and the Should Haves (the expected case) at the agreed time, then the estimating process is working well enough, the team is working effectively, the delivery plan is being validated and the risks are being managed. This Timebox-level confidence feeds upwards to instil confidence at the Project Increment level and the project level.

14. People, Teams and Interactions

14.1 Introduction

Collaboration and good communication are major factors in ensuring successful projects. DSDM places such importance on these that two of the eight principles highlight these topics.

Principle 3 – Collaborate

Principle 7 – Communicate continuously and clearly

Poor communication is recognised as the major cause of project failure. There are many examples of poor communication on projects; these are just some of the more common ones:

- Not defining and using shared language
- No access to the right people at the right time, particularly lack of verbal contact
- Ignoring information which is difficult to deal with or which contradicts your own view
- Key stakeholders not being identified or kept fully informed
- Appropriate communication channels not considered
- Only communicating using the written word

Poor communication on a project often results in delivery of the wrong solution.

DSDM directly addresses a number of such issues which improve communication and collaboration.

14.2 Effective Communication

14.2.1 Communication skills and the use of terminology

Effective communication skills are essential for all those involved in a DSDM project. They underpin many of the DSDM principles and are vital for effective team-working and for ensuring the transparency that all Agile approaches rely on. It is also important to remember that face-to-face communication is as much about listening (verbal messages) and watching (body language messages) as it is about speaking. In some circumstances a lot of information can be gained without saying a word; and like many of the soft skills, communication skills are not easy to teach or to learn. They can be worked on and improved, but they rely on a basic willingness and desire to improve communication in the first place. If this desire is absent, then communication is not two-way, but ends up more as a question-and-answer session, driven by one side or, worse still a documented statement 'sent out' for feedback and approval. This is not the DSDM way, as DSDM fully supports the Agile manifesto statements:

> *"We value: Individuals and interactions above processes and tools"*
> *and "Customer collaboration over contract negotiation"*

The soft skills possessed by the team, and the importance of those skills, should not be underestimated. When recruiting team members, selection criteria should balance technical capability and soft skills where possible. A well-performing communicative team can achieve far more than a team with the same technical capabilities but without these soft skills.

Teams may benefit from some short sessions to explore the way they communicate or possibly from some behavioural coaching. In any case, the way in which the team communicates and the effectiveness of their communication, both internally and with those outside the team, should be a key part of the retrospectives held throughout the project, evaluating soft skills such as:

- Listening
- Cooperation, compromise, negotiation
- Open, honest, transparent interactions
- Non-verbal communication, e.g. body language, tone of voice etc.
- Self-awareness
- Appreciation of others, empathy

It is also important to recognise that building a DSDM team brings together people who often use a different vocabulary or specialist language, based on their area of expertise. Members of the team should always try to use plain shared language, where possible. Where this is not possible, then it is good practice to maintain a glossary to define each term and its agreed definition. This simple step can avoid much confusion.

14.2.2 Planning effective communication

Effective communication does not happen automatically, especially now there are many channels of communication to choose from. One of the key risks is that starting to rely too heavily on technology for communication stops people from communicating effectively. Most people have experienced situations where heavy use of email starts replacing person-to-person conversation, resulting in the loss of much of the real meaning:

- Emailing the person at the next desk! This may give the perception of not disturbing them but misunderstandings and lack of clarification often cause problems later on
  - Some organisations address this with guidelines for email etiquette
- Being inundated with high volumes of emails, many of which are copied to a large number of people (just in case). This can result in important issues being missed

Communicating effectively actually requires some careful thought on how to take full advantage of the communication options available. It is important to identify which style of communication will give the best results in the particular circumstances and to select the most appropriate style for the situation. It is worth lightly considering such aspects as the different communities, the type and frequency of communication, the best medium and which roles are best suited to communicate with them. For example, a solution may be delivering a significant change to a user community and such a message is usually best delivered face- to face (if possible) by the Business Visionary, or possibly the Business Ambassador.

Well-known techniques such as stakeholder analysis and stakeholder management are also very useful to support decisions about communication. For example, identifying key stakeholders (in terms of power/influence and interest) who warrant face-to-face communication, often on a one-to-one basis. There will also be a number of people with little power or influence, who simply need to be "kept in the loop". For these people, simple email communication may genuinely be the most effective choice. The important thing is that such decisions are informed choices, and that effective communication is a two-way process.

Even a very light and informal assessment of the most appropriate method, frequency and process for different communications helps to focus communication. It is also important to be clear which decisions need to be recorded, to ensure everyone is clear which decisions have been made, and also to ensure key decisions are communicated.

14.2.3 Communication choices

1) Face-to-Face: Face-to-face communication will usually be the most effective way of communicating, either with individuals or with a small-to-medium sized group (as a workshop or a meeting). As well as enabling the rationale behind decisions to be fully understood, face-to-face communication allows immediate clarification of misunderstandings.
   - Allows communication using all the senses, but particularly words, tone of voice and body language

2) Conferencing - Video Conference (VC) and its variations: If those involved cannot physically get together, then a video conference is often the next most effective channel. For preference, each local group should gather in the same room, sharing one link in to the group conversation.
   - VC - Allows communication using words, tone of voice and body language (although body language communication may be limited by the field of vision of the video link e.g. may be facial expression only as well as the performance of the link which, where poor, may lead to 'blocky' images or a lag between sound and vision)
   - Teleconference – Allows communication using words and tone of voice, but excludes body language

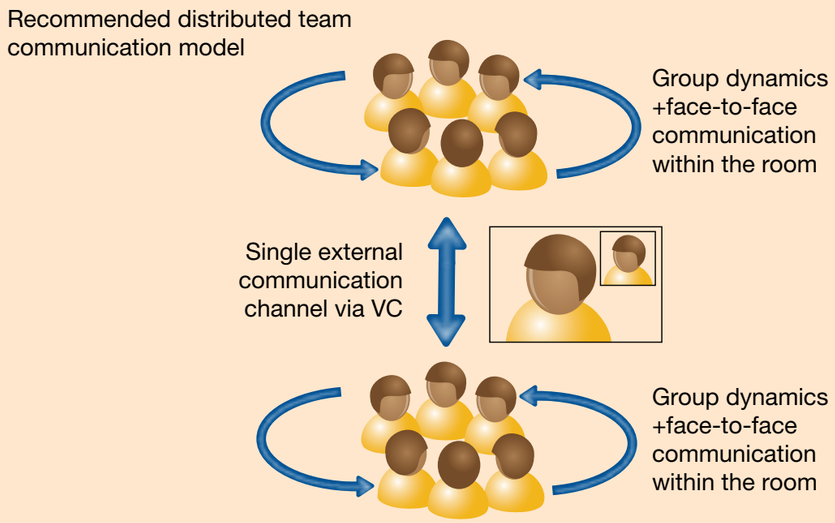


Figure 14a: Distributed team communication at its most effective

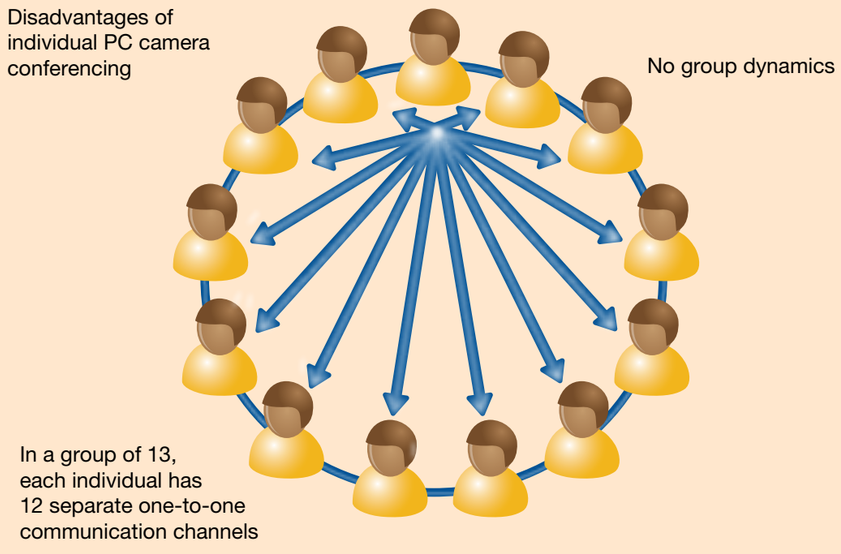


Figure 14b: Distributed team communication at its least effective

For preference, distributed teams should run their Daily Stand-ups around their Team Board with some kind of video link to the remote team members. Some DSDM teams will even have a live link running between onshore and offshore team members in order to simulate joint workplace. However, if these options are not possible then individuals can use headsets and video cameras on their PCs, although this is less effective than communicating as local groups.

However, where there is a large group in one location and several smaller groups or individuals all in separate locations, it may be more effective (and fairer) for everyone to dial in or VC so that everybody is working with similar constraints as regards to making themselves heard. Alternatively, consider a more facilitated session where somebody in the central location is responsible for ensuring the remote participants can participate effectively.

Whichever version of conference is chosen (video or telephone), everyone should aim to make each session as dynamic, vibrant and fun as possible, as if all team members were together. This can be done with a little thought, and it will greatly increase the real atmosphere of the DSDM style of working.

3) Chat facilities: For quick interchange of short pieces of information, this can be very effective. The sender can usually see whether the recipient is actually online and so they can expect a fast response. Most chat facilities have an automatic record feature, allowing the conversation to be saved.
   - Allows communication using words
   - Excludes tone of voice and body language

4) Email: Often treated (wrongly!) as the default communication channel. Email can be very effective for confirming what has been previously agreed (where the earlier discussion has used a communication channel higher up on this list). Email is also an effective channel for broadcasting information to a large group where getting people together is not justifiable, provided email best practice is followed.
   - Allows communication using words
   - Excludes tone of voice and body language

   To be most effective, emails should remove as much of the unnecessary "padding" as is possible whilst still ensuring the message is clear and stands out. Care should be given to elements such as the email title (so that the recipient can quickly assess how relevant the content is to them) and the list of To: and Copy: recipients.

5) Collaborative workspaces: These can be very effective for communicating informally within a team.
   - Allows communication using words, models, pictures

   Some workspace examples are project websites, intranets/extranets, Googledocs, Dropbox, Huddle; there are many others.

6) Documents: These are still important in DSDM, for capturing and managing more formal information and artefacts which need to be shared and managed, for example a risk log. However, where information is volatile or needs clarification, a document would normally be the last choice for effective collaborative communication.

A useful guideline for creating documents is "If you don't know who is going to read the document, then don't write it. And if you do know, ask them what they need it to contain".

Where possible, documents may communicate more effectively when pictures, models and diagrams are included, in preference to relying solely on large blocks of solid text.

- Allows communication using words, models, pictures.

14.2.4 Words or pictures?

Figure 14c: "A picture is worth a thousand words"

Sometimes the most effective communication can be by drawing by a simple picture, rather than giving a verbal or written description. DSDM actively encourages the use of visualisation through the use of practices such as Modelling and Iterative Development: both practices enhance the effectiveness of the spoken or written word.

14.2.5 Ongoing day-to-day communication

Similarly to all other Agile approaches, DSDM chooses from all available options to share information in the most effective way.

14.2.5.1 Team Boards

These are also sometimes referred to as Information Radiators (IRs), Kanban Boards or Big Visible Charts (BVCs).

They are simple, easily understood, often graphical representations of key information about the project. Although they are normally used for the current Timebox, they can also be useful at project or even at programme level. Team Boards form an intrinsic part of information sharing. For a co-located team, these are usually held on a physical wall or white board. However for a distributed team, they will have to be electronic (or both physical and electronic, if this is practical).

An effective Team Board provides a summary of team progress and the current status of their work. Ideally it should never be more than one working day out of date. It forms the focus for the Daily Stand-up.

Team Boards enable both the team members themselves and also any interested stakeholders to be able to see for themselves progress and other information with minimal effort. This removes the need to interrupt team members simply to find out "What's going on".

At its simplest, for a Timebox for example, the Team Board should show

- The objective for this Timebox
- The requirements or Stories that the team have committed to deliver, and which of these stories are not started, which are in progress, and which are "done". This information is usually shown by the position of the story (and potentially its associated tasks) on the Team Board. Displaying information in this way is based on the concepts of Kanban
- Any work that is currently blocked (the use of bright warning-type colours is an effective way of emphasising the importance and risk of these blocked items)
- A countdown of the time left to the end of this Timebox (to emphasise the principle of deliver on time and to keep the team focused)
- Any significant risks or issues for this Timebox and the risk owners

This simple level of information may be expanded to provide addition information. However, it is important that the Team Board does not become so cluttered that it loses clarity.

Team Boards can also be used very effectively in the wider organisation, whether for tracking the progress of specific activities, or for showing the current state and progress of programmes of work.

14.2.5.2 Daily Stand-ups

The Daily Stand-up is the opportunity for the team quickly to catch up on where everyone is; to make any fine adjustments to the plan for the next 24 hours, and to flag up very early if any significant problems are starting to appear and to re-affirm that, as a team, they are still on track to deliver what they agreed to deliver at the end of the current Timebox. This session may either be facilitated by the Team Leader, or preferably run by the team members themselves as a self-organising Solution Development Team.

Full details of the Daily Stand-up can be found in Chapter 13 – Timeboxing.

As part of (or following on from) the Daily Stand-up, the empowered team may decide to swap tasks around, to take advantage of a task that has finished early or to reduce the risk of a task that is taking significantly longer than expected. This demonstrates not only effective communication, but collaboration in action.

14.2.6 Co-located teams

The ideal situation for effective communication (and collaboration) is where the team is co-located. It is always easiest to share information when other team members are sitting nearby. Wherever possible, as a minimum, the Solution Development Team should have their desks grouped together, and ideally include one or two spare desks for business roles as and when they spend time with the team. It is also useful to be near to, or to have access to an informal area for Stand-ups or wider team conversations to avoid disturbing others who may be engrossed in their own work at that time. However, such ideal conditions are not always possible or practical, since teams may be working split across different floors, sites, towns, countries, continents or time zones.

14.2.7 Distributed teams

In some circumstances, co-location is simply not possible or practical. Where people are based in different locations on one or more sites, then careful consideration needs to be given to methods of effective communication and how communication will work. Effective communication requires effort and planning, it does not just happen automatically with no effort.

There is already a lot of valuable guidance published on distributed team working, especially in relation to off-shoring. However, in reality the problems are similar whether the site is split across continents or just at opposite ends of a city or even on different floors of the same building. It is just that cross-continent working adds more complexity around time zones and language and cultural differences!

Despite the technology available to support distance communication, if possible, a team should plan in some early face-to-face sessions to help establish a good working relationship. This in turn ensures that video-conference or teleconference sessions are more effective.

14.3 Collaboration

14.3.1 What is collaboration?

Collaboration is defined as "The action of working with someone to produce something".

Collaboration is first and foremost about people. A good starting point for collaboration is the working relationships. Where there are healthy working relationships, this is where the best collaboration takes place. True collaboration, especially in a team context, is about ensuring give and take on all sides and being comfortable with this.

There are also a number of important cultural factors to consider for effective collaboration, (for example, values, beliefs and assumptions) and around personal style (personalities and behavioural preferences).

In particular, individual characteristics have a direct influence on the ability to collaborate. Those who are approachable, personable, good at forming relationships, open and good natured are typically, naturally collaborative. For others, improving collaborative behaviours may require some conscious effort or coaching. Some people prefer working in a collaborative way, for others, a collaborative style of working would not be their first choice. In order to foster a collaborative DSDM culture, it is important to recognise and accept that this style of working comes more easily to some than others and to provide support and encouragement for people and teams as they start adopt a different (DSDM) style of working.

Collaboration is about understanding and working with differences of opinion and differing views. There is demonstrable value in fostering a collaborative team culture, both in terms of building better solutions, but also in terms of motivation and job satisfaction for individuals.

14.3.2 Building effective collaboration

Certain ingredients are needed to build effective collaborative teams. Some examples of these ingredients are:

- Having mutual trust between team members
- Having mutual respect between team members
- Being open-minded as an individual
- Being approachable as an individual
- Being available when needed
- Being open to change
- Having a clear direction
- Having a consistent and stable team membership
- Having belief in yourself as an individual and belief in others
- Communicating properly and effectively (e.g. using active listening, empathy etc.)
- Being subject to effective (DSDM-style) leadership

It is very rare to have the perfect team and the perfect environment already in place. So in reality, some of these ingredients may already be in place, others may need work to improve or establish them.

In order to capitalise on the value that collaborative working brings, it is sometimes necessary to address issues within the organisation or within individuals that act as barriers to collaboration.

Some examples of these collaboration barriers are:

- Organisational structures which group similar types of skill in silos, encouraging and supporting a "them and us" culture
- Managers who have to resource their projects based around scarce resources
- Organisational focus and reward based only on personal, individual goals
- Unwillingness to concede something for "the greater good" which would impact a personal goal
- A culture of personal competition

Collaboration can sometimes find itself in direct conflict to competition. In some organisations and cultures, recognition and reward, e.g. appraisals and salaries, are based on competition and being better than others, rather than around team achievement. As a result, this encourages individual success (competition) rather than team success (collaboration). These barriers need to be addressed to foster an environment where collaboration can thrive.

14.3.3 Collaborative people

Collaboration is all about problem-solving, bringing together people with expertise in very different areas in order to find a solution to the problem. Effective collaboration works best where individuals possess T-shaped skills: on the one hand, possessing a deep knowledge of their own discipline (the vertical part of the T), on the other hand, understanding how their discipline interacts with others (the horizontal part of the T).

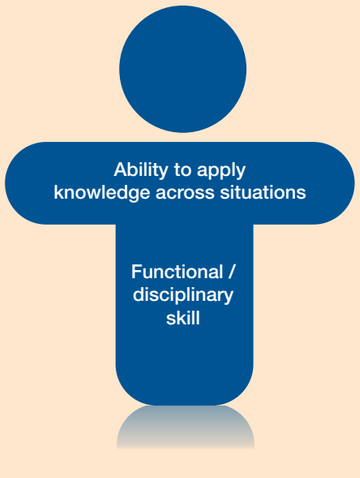


Figure 14d: T shaped skills

Having this breadth of experience is useful in dealing with both opportunities and problems. People with this broader skillset can apply different perspectives to their thinking to help the team form a cohesive approach. An understanding of how an individual's discipline interacts with others is vital to help individual team members build a collaborative team.

14.3.4 DSDM teams

All Agile approaches place a high emphasis on the concept of "team". DSDM teams bring together people from the business and solution community to work jointly towards a single shared goal. This is a very different scenario from many traditional style projects, where the divisions, the "us and them" culture, is reinforced from day one. However, unlike most other Agile approaches, DSDM takes a wider view of the concept of "team", recognising the importance of engaging the wider stakeholders. So as well as the Solution Development Team roles, DSDM identifies the roles that are focused on the overall project, and the roles that are supporting the Solution Development Team. DSDM also strongly recommends ensuring that the standard best practice on stakeholder identification and stakeholder management should form part of the lifecycle, particularly Feasibility and Foundations activities. By ensuring everyone has the full picture of those involved in the project, those impacted by the project and those who will have an impact on the project, DSDM minimises the risks of nasty surprises later in the project.

This chapter focuses predominantly on collaboration for the project (based on the roles as defined in Chapter 7). However the guidance can also be applied to any external stakeholders for the project.

14.3.5 Shared team goals and ways of working

The cornerstone of team collaboration is having a single shared goal and ensuring this goal is visible to the whole team. Without a single goal, which the whole team buys into, collaboration is at risk as competing goals or personal agendas emerge.

It is also very beneficial to agree the team "norms" - the behaviours that form the basis of the team's interactions and how they expect to work together.

As part of the lifecycle, the early phases of a DSDM project and the early DSDM products are important in bringing people together and making sure that common goals are established and visible to all. Products such as the Business Case, which define and publish the business vision ("a clear and concise statement of where the business or product expect to be after the project has completed") are key to ensuring everyone on the team has the same understanding of the reason for the work and the value it will deliver to the business. This simple step alone removes much potential confusion. For many teams unused to a DSDM style of working, seeing this information stated in clear language, emphasises DSDM's strong focus on effective communication as well as being a motivator to do a good job.

14.3.6 A culture to support collaboration

Collaboration is usually most effective where there is a supportive culture in place. This provides members of the team with the confidence and trust to be open and honest, and thus paving the way to ensure issues are raised early. When issues are raised early, solutions can be found while there are still options available. The later an issue is raised, the less choice is open to the team.

A blame culture is the antithesis of a supportive culture. Where a blame culture exists, it prevents the necessary honesty, and often results in behaviours such as:

- Spending time and effort shifting responsibility or preparing a defence in case problems arise in the future, rather than simply sorting the problem out now
- Hiding a problem, in the hope that it will somehow be resolved before anyone finds out
- Over-estimating the time needed for tasks, to avoid the perception of "failure"

A supportive culture means that there is recognition that mistakes do happen but that lessons are learnt and the team ensures that they are not repeated.

The DSDM practices of Workshops (Chapter 9) and Iterative Development (Chapter 11) also help to build and support a collaborative culture.

14.3.7 Leadership in a collaborative culture

A collaborative culture requires a particular style of leadership – facilitative and collaborative (as opposed to the authoritarian "command and control" style so often encountered on traditional-style projects). This facilitative and collaborative leadership style is promoted by all Agile approaches, and is sometimes referred to as "servant-leader" . However as a "servant-leader", there is still an expectation that the leader will be an active participant, rather than a passive bystander, while always encouraging the team to be self-organising.

In particular, the DSDM roles of Project Manager and Team Leader should actively encourage, promote and support a no-blame culture and where necessary protect the team from external pressure, although all roles share the responsibility for making this happen. See Chapter 7 – Roles and Responsibilities for more detail of the style expected from Project Manager and Team Leader.

14.4 Summary

In summary, the importance of effective communication and collaboration cannot be underestimated. It is directly linked to the whole ethos of DSDM working. DSDM came about directly because technology allowed the creators of a solution to sit alongside the requestors of the solution and to talk to one another, providing short communication loops to evolve the solution. Initially this was often on a one-to-one basis and occurred where the creators self-selected and actively chose to work this way. But as DSDM has grown over the years, as the teams scale up, as teams involve people who have not selected themselves for this style of working, then everyone needs to remind themselves of the basics of effective communication and collaboration, since it is only when communication and collaboration are really working well, that a DSDM project can succeed.

15. Requirements and User Stories

15.1 Introduction

The importance of a well understood, prioritised and agreed set of requirements is self-evident. However, the attempt to define a full and detailed set of requirements too early in a project often proves to be counter-productive, restrictive and wasteful. It is not possible to define all of the detailed requirements at the outset of a long project. The business environment changes as time progresses; new requirements and opportunities present themselves. As the project progresses, the team understand more about the business need. Defining detailed requirements too early means either needing to change the specification later, which wastes the original work, or delivering to the originally-specified requirements and subsequently failing to adequately satisfy the business need.

DSDM acknowledges this dilemma and proposes a better way of working. DSDM advises the capture of requirements at a high level, early in the project. Further detail is gradually elicited as the project progresses, deliberately leaving the finer details as late as practicable, i.e. until the Evolutionary Development and the Timeboxes.

15.2 What is a Requirement?

At its simplest, a requirement is a service, function or feature that a user needs. Requirements can be functions, constraints, business rules or other elements that must be present to meet the need of the intended users.

For example:

in a training company with its own training centre:

- *The Course Manager has a requirement to schedule training courses and reserve rooms, in order to make available courses visible and to ensure courses run effectively*
- *The Training Centre Manager has a requirement to keep track of what training is running, in order to ensure appropriate allocation of trainers to courses*
- *The Financial Accountant has a requirement to maximise the amount of time that the training rooms are in use, in order to maximise revenue from the rooms*

If the product to be delivered is a custom-built car, the requirements defining this would be more feature-based:

√ *A means of propulsion*
√ *A maintainable steering capability*
√ *A comfortable place to sit*

However, it should be noted that the following are not requirements, but solutions:

X *An engine*
X *A steering wheel*
X *Bucket seats*

DSDM projects aim to state requirements in a manner which avoids tying them to a particular solution for as long as possible. This is because more flexibility can be retained in how a solution is eventually provided if requirements are expressed as what needs to be achieved, rather than how they will be met from a technical point of view, e.g. *"a means of propulsion"*, rather than *"an engine"*. A solution expressed too early may become a constraint on what can be achieved within time and budget.

15.2.1 Categories of requirement

The success of any solution is the product of two aspects:

- what it does (functionality, features)
- how well it performs against defined parameters (non-functional attributes, acceptance criteria, service levels)

15.2.1.1 Functional Requirements (FRs)

Functional requirements express function or feature and define ***what*** is required, e.g.

- *Visit customer site*
- *Obtain conference venue*

The requirements do not state ***how*** a solution will be physically achieved.

- *Drive to customer site* is one possible solution. However, *fly to customer site or travel by train to customer site* are potential alternative solutions which may be worth consideration
- *Build conference centre* is one possible solution. *Hire a hotel room* is an alternative solution

Stating requirements early in the project as ***what*** rather than ***how*** allows room for flexibility and innovation later.

15.2.1.2 Non-functional Requirements (NFRs)

Non-functional Requirements define ***how well,*** or to what level a solution needs to behave. They describe solution attributes such as security, reliability, maintainability, availability (and many other "...ilities"), performance and response time, e.g.

- *responding within 2 seconds*
- *being available 24 hours per day, every day*

These NFRs may be:

- Solution-wide or impacting a group of functional requirements: e.g.
  - *All customer facing functionality must carry the company logo*
  - *All customer-facing functionality must respond within 2 seconds to requests*
- Related to a particular functional requirement, e.g.
  - *Hire conference venue might have NFRs of accessibility, security, and availability*

15.3 User Stories

15.3.1 What is a User Story?

A User Story is a requirement expressed from the perspective of an end-user goal. User Stories may also be referred to as Epics, Themes or features but all follow the same format.

A User Story is really just a well-expressed requirement. The User Story format has become the most popular way of expressing requirements in Agile for a number of reasons:

- It focuses on the viewpoint of a role who will use or be impacted by the solution
- It defines the requirement in language that has meaning for that role
- It helps to clarify the true reason for the requirement
- It helps to define high level requirements without necessarily going into low level detail too early

User goals are identified and the business value of each requirement is immediately considered within the user story.

User Stories are often deemed to comprise three elements - **the 3C's**

- **C**ard
- **C**onversation
- **C**onfirmation

15.3.2 User Story format

The format of the User Story is as follows:

As a *< role>*

I need *<requirement or feature>*

So that *<goal / value>*

These two examples demonstrate User Stories at different levels, but using the same format:

At a project level

As a *Marketing Director,*

I need *to improve customer service*

So that *we retain our customers.*

At a detailed level

As an *Investor,*

I need *to see a summary of my investment accounts,*

So that *I can decide where to focus my attention.*

User Stories provide another powerful message. Choosing User Stories to define requirements demonstrates an intention to work collaboratively with the users to discover what they really need. The User Story is brief and intended to be a placeholder for a more detailed discussion later – the *Conversation*. Much of the detail of User Stories emerges during Timeboxes as part of evolutionary development. High-level User Stories (Epics) are broken down by the Solution Development Team into more detailed User Stories just before development commences on that group of stories. Even then, the User Stories are not intended to be full specifications of the requirements. Fine detail may not need to be written down at all, but may simply be incorporated directly into the solution as part of the work within a Timebox.

The user story format helps to ensure that each requirement is captured in a feature-oriented, value oriented way, rather than a solution oriented way.

In DSDM projects, User Stories are recorded in the Prioritised Requirements List (PRL).

This is the equivalent of a Product Backlog in other Agile approaches.

15.3.3 User Story – the *Card*

From the PRL, User Stories are often printed onto physical cards, for planning purposes and to help the Solution Development Team monitor progress.

The Front of the Card

On the front of the card, the following information is typically displayed:

- A unique "Story Identifier", usually a number or reference
- A clear, explicit, short name or title
- "As a <user role> I need <requirement>, so that <business reason/value>";

  This section states:
  - who is the primary stakeholder (the role that derives business benefit from the story)
  - what effect the stakeholder wants the story to have
  - what business value the stakeholder will derive from this effect.

The Back of the Card

On the back, the *Confirmation* area contains:

- Acceptance criteria (the test criteria)

These acceptance criteria define, at a high level, the test criteria which will confirm that this user story is working as required. These are not intended to be the full test scripts, but will be used to expand into the appropriate test scenarios and test scripts during Timeboxes, as necessary.

For User Stories at the highest level (sometimes called a project Epic), the acceptance criteria may be used to define the aims of the project using criteria that may be measured after the project has completed (as part of the Benefits Assessment).

Project acceptance criteria example:

- Is customer retention improved by 20% within two years?
- Is product range increased by 10% within 5 years?
- Has speed of dispatch improved to within 24 hours of time of order for 99% of in-stock items within 6 months?

User Story Example:

Story Identifier: STK001

Story Name: Customer Order

Description: As a Customer, I need to place an order so that I can have food delivered to my house.

Confirmation: Acceptance Criteria examples:

Functional:
- Can I save my order and come back to it later?
- Can I change my order before I pay for it?
- Can I see a running total of the cost of what I have chosen so far?

Non-functional: availability:
- Can I place an order at any time (24 hours per day or 24/7/365)?
- Can I view the order at any time (24 hours per day or 24/7/365) up to and including delivery?

Non-functional: security:
- Are unauthorised persons and other customers prevented from viewing my order?

15.3.4 Well constructed User Stories

Bill Wake's INVEST model provides guidance on creating effective User Stories:

Independent	Stories should be as independent as possible from other stories, to allow them to be moved around with minimal impact and potentially to be implemented independently. If stories are tightly dependent, consider combining them into a single user story.
Negotiable	Stories are not a contract. They are "placeholders" for features which the team will discuss and clarify near to the time of development.
Valuable	Stories should represent features providing clear business value to the user/owner of the solution and should be written in appropriate language. They should be features, not tasks.
Estimable	Stories need to be clear enough to estimate (for the appropriate timeframe), without being too detailed.
Small	Stories should be small enough to be estimated. Larger "Epic" stories should be broken down into smaller User Stories as the project progresses. The stories after splitting still follow the INVEST criteria.
Testable	Stories need to be worded clearly and specifically enough to be testable.

A well-written user story is clear, concise and complete. Some simple checks are:

- It does not combine with, overlap nor conflict with other User Stories
- It conforms to organisational and project standards and policies where applicable
- It is traceable back to the business needs expressed in the business case and project objectives
- Where several User Stories relate to the same feature, but for different users, they are cross-referenced to each other

15.4 Requirements Through the DSDM Lifecycle

Projects need:

- A clear project objective
- A statement of the business vision
- A Business Case, agreed with key stakeholders

These form the anchor for the deliberate evolution of the more detailed requirements, iteratively and incrementally, as the project progresses. As the hierarchy of requirements emerges in expanding detail, as the project unfolds, each requirement/User Story can always be traced back to this original vision, as it evolves to meet the real and current business needs.

15.4.1 Requirements activity during Feasibility

All projects begin with an idea and an expectation of benefits to give a return on investment. The Business Analyst ensures that the Terms of Reference (which is sometimes vague or unclear) is expanded to provide a clear project objective, a business vision and an outline Business Case. The project vision is clarified and key project objectives are defined. The highest level Epic User Story is the objective of the project. The User Story format can be effectively used to clarify:

- Who needs this? (Do we have the right Business Sponsor?)
- Why do they need it? (What is the key business value expected or needed?)
- What are their expectations? (What are the high-level acceptance criteria?)

The User Story format also helps to identify the key stakeholders with whom to gain agreement for the requirements.

In Feasibility, the User Stories (sometimes called Epics or Themes) should constitute a small number of clear statements that are just sufficient to scope the project, to identify whether it is worth proceeding further and to establish likely costs and benefits achievable. DSDM recommends typically less than 10 requirements/User Stories at this point.

Non-functional requirements (see above) may also emerge at this point, but these are expected to become clearer and more detailed throughout the project. Some of the more critical ones may be evident from the outset, when the project objective is established, and these need to be captured because they may constrain some of the choices for the project.

Even at this high level, User Stories help to focus on the value of what is required.

> *For example:*
>
> *"As a Human Resources Manager, I need a better way to deal with employee records, so that employee history can be tracked including their training and career moves."*

is a far more effective way of defining what the business needs, than the vague but technically constraining statement:

> *"The organisation will implement a human resources system."*

The user story format helps to bring out the real objectives of a major change.

15.4.2 Requirements activity during Foundations

During Foundations, more understanding of the requirements is needed, sufficient to clarify the scope of the project, prioritise, estimate and formulate a realistic Delivery Plan.

During Foundations, the high-level Epic or Theme stories established in Feasibility are now broken out into simple User Stories (functional and non-functional). User Stories defined by the end of Foundations in a DSDM project must be specific enough to estimate and small enough to fit into a Timebox (typically 2 – 4 weeks work). This is not the lowest level of breakdown that the project will achieve, but by the end of Foundations User Stories need to be just sufficient to allow for estimates of work to be done and to plan a schedule of Timeboxes for the first Project Increment.

At Foundations, User Stories are assembled into a Prioritised Requirements List (PRL). The focus is on describing the business need embodied in each User Story, in a way which does not constrain unnecessarily how the requirement will be achieved.

Key non-functional requirements should also be considered and documented during Foundations. It may be difficult or impossible to accommodate such requirements if they are discovered too late in the project.

The PRL is baselined at Foundations, to give a clear checkpoint for the set of requirements which was used for planning. In this way, new requirements which emerge during development are clearly identified, and their impact can be assessed.

15.4.3 Requirements activity during Evolutionary Development

At the outset of each Timebox, the User Stories allocated to that Timebox will be further investigated. The User Stories from the PRL are broken down into more detailed User Stories which are small and clear enough for the team to work from. The detail is only elaborated one Timebox at a time, and thus the complexity of the requirements is managed. Also, the fine detail is only elicited immediately before that element of the solution is created. This avoids time being wasted on developing detail on all areas up front.

During Timeboxes, the detailed requirements/User Stories emerge iteratively. The Business Analyst captures the appropriate level of emerging detail within the PRL, where this is not adequately captured within test criteria, prototypes and the Evolving Solution itself. The Business Analyst also works collaboratively with both Solution Development Team and project level roles to help retain the project's focus on value and priorities.

New requirements may emerge which were not identified during Foundations. The Business Analyst facilitates the consideration and impact analysis of these and records their inclusion or otherwise in the project, based on discussions with the Business Ambassador, the rest of the Solution Development Team and/or Business Visionary. The Business Analyst also records details of, and reasons for, any lower priority requirements being de-scoped by team agreement during Evolutionary Development.

15.5 Conclusion

Requirements evolve and emerge in a DSDM project. Analysis of the detailed requirements is deliberately left as late as is sensible, to avoid unnecessary rework and to manage complexity. Because of this, it is important to obtain agreement to a high-level baselined set of prioritised requirements in the PRL in the early phases of a DSDM project. This gives scope, direction and an appropriate degree of control for the project to evolve the detail whilst allowing change to be embraced and controlled.

16. Project Planning and Control

16.1 Introduction

In common with all other Agile methods DSDM values *responding to change over following a plan* but, unlike some, it puts a greater emphasis on planning, specifically *high-level* planning, i.e. DSDMs plans shape and structure the project and the work but do not get into the detail of exactly who does what and when.

Planning in DSDM starts with agreement of strategy – the approach to be taken to evolve the solution – and considers:

- Incremental delivery of the solution – in increments and timeboxes
- Quality Assurance of the solution - how review and testing activity will be integrated into development

Strategic planning starts in the Feasibility phase with an initial outline of the project management and solution development approaches supported by a possible time-line for delivery. Assuming a preliminary judgement that the project is feasible, these outlines are gradually refined throughout the Foundations phase to generate high-level plans and agreed ways of working that form the basis of the commitment to delivering what the business needs when it needs it.

The following sections consider:

- What planning activity should take place at what phase in the lifecycle
- When agreements and commitments are made (incorporating formal sign-offs if appropriate)
- How to measure progress against plans
- How changes to those plan are handled in an Agile way

Plans always need to evolve to meet changes in real-world circumstances. This may be as a result of shifting business needs. Alternatively, as requirement and solution detail emerges over time, the understanding of what is possible and what is needed to deliver a valuable business outcome may change. Before that though, it is worth considering best practice concepts in project planning, testing, tracking progress and demonstrating control and how these are integrated with the DSDM Agile Project Framework

16.2 Project Planning Concepts

16.2.1 Outcome-based planning

A framework of empowerment exists in the hierarchy of a DSDM project. At the highest level, the Business Sponsor has empowered project-level roles to manage the delivery of a valuable business solution that will provide the expected Return on Investment. Below that, the project-level roles have empowered Solution Development Teams who self-organise to deliver the solution envisioned by the Business Visionary and Technical Coordinator to meet the business need.

Within this framework of empowerment:

- The Project Manager is responsible for high-level planning for the project, collaboratively planning for the incremental delivery of the business solution – the outcome required by the Business Sponsor
- The Solution Development Team is responsible for planning the detail of all the work of each Timebox with team members agreeing amongst themselves who will do what work to achieve the objectives agreed at the Kick-off of that Timebox

16.2.2 Planning to sensible horizons at the right level of detail

The planning horizon defines the period of time to be covered by a plan. The two plans defined by DSDM cover two very different planning horizons with the Delivery Plan looking to a horizon of the end of the project – which will probably be months or sometimes even years in the future – and the Timebox Plan looking to a horizon of the end of a Timebox – typically no more than 4 weeks into the future. The use of planning horizons, in conjunction with the consideration of an appropriate level of detail, is intended to deal with the issue of uncertainty. Common sense says that planning activity for the next few hours to quite a fine level of detail is useful but that the value of detailed planning reduces as uncertainty increases and the longer the timeframe, the more uncertain the outcome.

For example – planning a journey

Planning a journey to a particular location and setting off in the next hour or two could sensibly involve exact bus and train timings, precise predictions of how long it takes to walk to the bus stop etc. An accurate knowledge of the environment and the current circumstances allow precision and realism.

Planning a similar journey setting off 6 months from now could not sensibly involve the same precision because there is less certainty of the environment at that time and no way realistically to predict circumstances. Will the bus and train timetables be the same as today or will one or both have changed to an as yet unspecified new timetable? Will it take as long to walk to the bus stop as it does today? (For example due to suffering from a sporting injury) And so on.

A typical Delivery Plan for a project will provide a schedule of Timeboxes and any other high-level activities for the imminent Project Increment – a planning horizon of perhaps 6 weeks to 6 months – and is likely to have objectives and perhaps delivery dates for future Project Increments – a more distant planning horizon significantly further into the future.

A typical Timebox Plan – with a much shorter planning horizon (typically 2-4 weeks) – is likely to be much more detailed – maybe even down to task-level detail of exactly who intends to do what and when.

When deciding on an appropriate level of detail for a given planning horizon it is important to balance the likelihood of change to the plan against the security of understanding in the detail of what needs to be done. There is no point in going into lots of detail now if that detail will not stand the test of time. Detailed plans tend to act as a barrier to change and it is important to remember the philosophy of DSDM is that *best business value emerges when projects are aligned to clear business goals, deliver frequently and involve the collaboration of motivated and empowered people*. Care needs to be taken to ensure that the level of detail in plans does not stifle the emergence of detail or undermine the empowered collaboration that supports the emergence of that detail.

16.2.3 Plan and re-plan based on best available estimates

Estimates evolve as more is understood about the thing being estimated. Early in a project, estimates will be uncertain and can only be expressed with a low confidence factor typically described by a wide range. For example, by the end of the Feasibility phase, a project may be estimated to require 1000-2000 days of effort to complete. By the end of the Foundations phase there is a need to be more precise with estimates as delivery dates and associated costs need to be committed at this point. A typical estimate at this point should be accurate to within a much narrower band than at the end of Feasibility. For example, for the same project as used in the previous example, a range of 1500-1800 days of effort to complete the project should hopefully be achievable. It is important to achieve a level of confidence with estimates that would allow for the Must Have requirements for the project to be guaranteed as part of the solution and to have a reasonable expectation that the Should Have requirements will also be delivered. The top end of the range should be set at a level where it is *possible* to deliver all the Could Haves as well but this should be acknowledged to be unlikely – the Could Haves represent the primary contingency for the project.

As the project proceeds and more becomes known about the requirements and the Evolving Solution, and the accuracy of previous estimates is validated by actual development work, it makes sense to evolve the Delivery Plan (the schedule of what requirements will be addressed in what Timebox) and to refine predictions of what will be delivered within the fixed timeframe of the current Project Increment and perhaps the project as a whole.

Regardless of the amount of investigation work that has been done, estimate accuracy can be improved by following two essential aspects of estimating best practice, specifically: estimating using more than one technique and estimating in groups.

Estimating using more than one technique

Early in the project estimating by analogy (or 'top-down' estimating) is typically the most sensible basis for estimating the effort needed to complete the project. The estimator uses knowledge of current requirements and development tools, techniques etc. and experience of doing similar work in the past as the basis for estimating by analogy.

The application of a second technique by the same estimator will help validate the first estimate. Estimating by decomposition (or 'bottom-up' estimating) is a common second technique to use. It requires a **sample of the requirements** to be taken – typically higher priority requirements likely to be addressed in an early Timebox. These requirements are then analysed in detail (probably to the level of detail described for the Investigation step of a structured Timebox) and broken down into tasks needed to complete the work on those requirements. The tasks associated with a requirement are estimated individually and those estimates are added together to provide an estimate for the requirement as a whole.

If there is a significant discrepancy between the estimates generated by the two techniques then further work will be needed to understand why and what the final estimate should be.

Estimating in groups

Regardless of the technique used for estimating, accuracy can be improved by collaboration. The commonly used Agile technique of *Planning Poker* combines a number of techniques into a simple and effective technique that involves the use of custom-design playing cards. In Planning Poker, classic Wide-band Delphi (where iterations of discussion and re-estimating bring the opinions of estimators closer together) and comparative estimating (where one requirement is estimated in terms of multipliers of a requirement that everybody agrees will take the least effort to complete) are brought together in a framework that acknowledges that the larger the thing being estimated, the less precise the estimate will be.

16.3 Testing Concepts

16.3.1 Testing integrated throughout

Early in a project, it is important to ensure that a strategy for testing is in place and that everybody understands their responsibilities with regards to solution quality and how this is assured by an appropriately rigorous regime of review and testing. Testing should be considered part of the Iterative Development process with testing activity as fully embedded as it can be within the same Timebox as the development activity. This is because the earlier a defect is found, the easier and cheaper it is to fix.

Ideally the solution will be fully tested and potentially deployable at the end of the Timebox, although it is acknowledged that this may not be achievable in all circumstances.

16.3.2 Collaborative testing

Effective and productive testing involves the collaboration of all stakeholders on the project to increase the productivity of the test-fix-and-retest cycle. This concept is in line with the DSDM principle to *Collaborate* and should include business and technical, solution development and testing representatives.

16.3.3 Repeatable testing

Since testing needs to support the DSDM principle to *Build Incrementally from firm foundations*, it is important that testing within a Timebox includes not only tests for the new features of the solution being built but where appropriate also tests for what has been built previously. It is therefore good practice to ensure tests are readily repeatable. Where appropriate, automation tools can be used to reduce the effort associated with repeating tests.

16.3.4 Prioritised testing

Although test automation tools may help reduce effort associated with repeated tests, there may still be a need to prioritise tests as there is not always time to exhaustively regression test all aspects of the solution as it evolves. In this circumstance, it may be helpful to prioritise testing on the basis of risk, and on the likelihood of having introduced a defect and the likely impact of such a defect. MoSCoW rules could be applied to both the execution of tests and the rectification of defects found. For example, when planning tests, decide: which tests Must be run; which Should be run; which Could be run; and specifically identify areas of the solution that Won't be tested this time. If defects are discovered, it can then be determined which of these:

- Must be rectified because the solution, or feature of it, could not be deployed with the defect unresolved
- Should be fixed because the impacts of ignoring the defect may be significant
- Could be fixed but the minimal impact in live use may be considered 'acceptable'
- Won't be fixed at this time because the impact of the defect is insignificant

16.3.5 Independent testing

A product should always be tested by someone other than its creator because it is as critical to test the understanding of a requirement as it is to test that the work done to fulfil that requirement was completed correctly. Even though individuals within a Solution Development Team may hold both Solution Developer and Solution Tester roles it is important that one individual always independently tests the work of another. Active involvement of the Business Ambassador and Business Advisor roles in the project always provides an independent perspective for testing.

16.3.6 Test-Driven Development

The concept of Test-Driven Development (TDD) turns traditional testing practice on its head. Traditionally tests are designed and built in parallel to the design and build of the solution with both being based on individual interpretation of a given requirement. This often leads to debate about whether the Developer's interpretation of the requirement is correct or whether the Tester's interpretation is correct. In some cases, a good test will correctly identify a defect in the solution, in others an incorrect test may suggest that a solution is defective when it isn't. Using a Test-Driven Development the design and build of the test precedes development of the solution and helps define the requirement. The solution, or feature of it, is then developed until it passes all the specified tests. Research has shown that the practice of Test-Driven Development significantly increases the overall quality of the solution.

16.4 Tracking and Control Concepts

At a high level, outcome-based measurement of progress ensures that tracking for the project aligns with the planning concepts of *outcome-based planning* and *planning to sensible horizons at an appropriate level of detail.* At the Timebox level, tracking is based on the concept of transparency of process and progress – effectively making visible what the team are doing and how they are progressing on a day-to-day basis. The value shared by all Agile approaches, which emphasises *responding to change over following a plan,* provides the main reason for tracking. It is important to understand the current position in a project so that the consequences of any new work required by the change can be assessed. Management by exception supports the framework of empowerment embraced by DSDM and the following concepts applied in combination allow the project participants to meet the eighth principle to *Demonstrate Control.*

16.4.1 Timeboxing and outcome-based measurement

The use of Timeboxes provides a structure of nested plans to support outcome-based measurement. Outcome-based measurement places the primary focus of measurement on what has been delivered as part of a Solution Increment at the end of a Timebox. The demonstration of the Solution Increment at the end of each Timebox and the formal acceptance by the Business Ambassador, or perhaps the Business Visionary that what has been delivered is fit for purpose provides the opportunity for outcome-based measurement.

At this level, understanding what has actually been delivered (the actual outcome) compared with what was planned provides the clearest possible indicator as to whether the Project Increment and ultimately the project as a whole is on track to deliver what was promised (the Must Have requirements) and what is expected (the Should Have requirements), as well as understanding what contingency for this remains (in terms of Could Have requirements). Discipline at the Timebox level is therefore the basis of control not only of the Timebox itself but also for the Project Increment and the project as a whole.

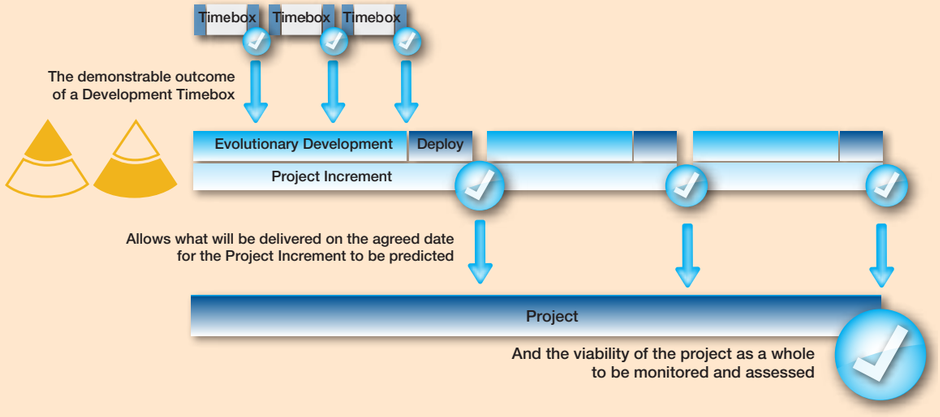


Diagram 16a - Increasing confidence

16.4.2 Transparency of process and progress

At the Timebox level, transparency of process and progress comes from the use of a Team Board and the Daily Stand-up that should take place near the Team Board. In combination these make visible the necessary elements of control at the level of the Solution Development Team.

The Team Board makes the detailed plan and activity against that plan visible to anybody who cares to look. The Team Board clearly shows who is doing what work to meet any particular requirement and, based on estimates of effort required to complete that work, whether the requirement is likely to be fulfilled and demonstrable in the Solution Increment. Issues are also noted on the Team Board along with ownership of that issue.

The Daily Stand-up, which involves everybody in the Solution Development Team from both the business and technical perspectives, provides an opportunity for each member of the team to describe:

- What they have done since the last Stand-up (describing progress)
- What they intend to do before the next Stand-up (planning in detail to a very close horizon)
- What, if anything, may be blocking their work (making issues visible)

Sharing information in this way provides opportunity for the collaborative, pro-active problem solving that characterises an effective Agile team.

16.4.3 Responding to change

In a dynamic business environment following an approach where the detail of understanding of the problem and the detail of what makes up the solution is expected to emerge over time, it is essential that change is not only accepted as inevitable but that it is welcomed as part of the process of getting the solution right. That said, it is equally important: to maintain a *focus on the business need* (Principle 1); to *deliver on time* (Principle 2); and *to never compromise quality* (Principle 4). This means that change should also be controlled.

Change control in a DSDM project tends to be more formal at the project level than it is at the Solution Development Team level.

At the project level, the Business Visionary is responsible for making sure that the solution meets the business vision and is expected to approve the high-level requirements, described in the Prioritised Requirements List, as a coherent set that reflects the needs and desires of the business. If, as development progresses, there is pressure to make changes to these high-level requirements then that change should be formally approved by the Business Visionary as being necessary and in line with the business vision. (This is sometimes referred to as a change in breadth.)

At the Solution Development Team level, most of the change will come as a result of a deepening understanding of a requirement or how that requirement will be fulfilled in the Evolving Solution. Change to depth and detail does not represent a formal change of scope and therefore it is primarily at the discretion of the team with the Business Ambassadors and Advisors empowered to decide what is appropriate and acceptable, within the constraints of time (and cost and quality) being fixed and requirements being negotiable.

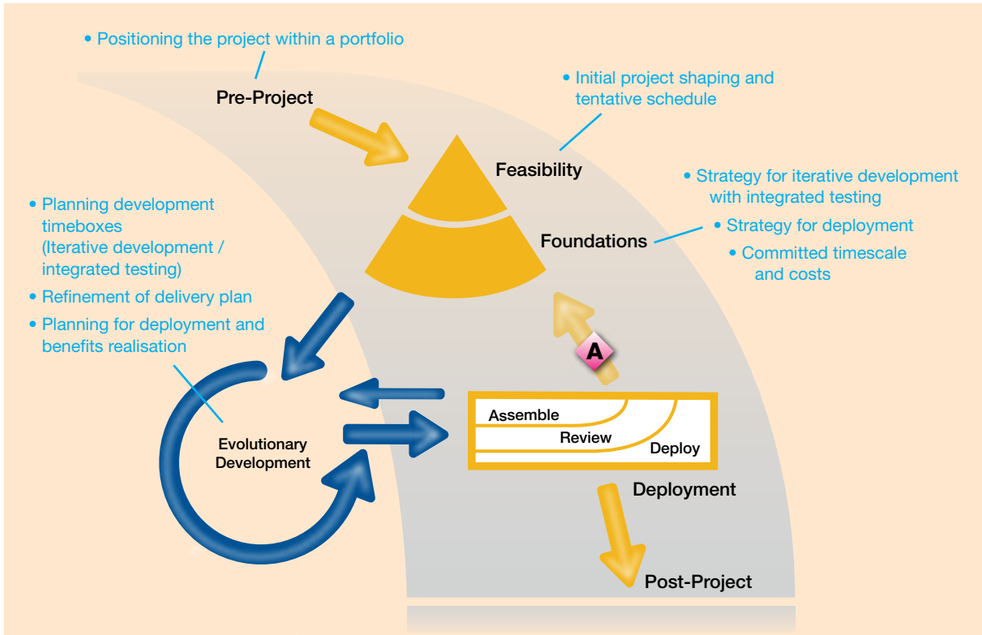
16.4.4 Management by exception

Within the framework of empowerment promoted by DSDM, and using the planning and control concepts described above, day-to-day management of the work required to evolve the solution is left to the Solution Development Team. A degree of ***tolerance*** related to the MoSCoW prioritised scope of what is expected to be achieved is built into the objectives for a Timebox. Typically, the Solution Development Team is empowered to de-scope any Could Have requirement without referring up to the project-level roles. Provided the team is confident that it can deliver a solution within this tolerance, it can make any decisions it needs to around the detail of what will be done and how. If however the team feel that the Solution Increment will not meet all the Must and Should Have requirements agreed or if meeting all the Must and Should Have requirements still risks compromising quality, then this is considered to be an ***exception***. Any exception should be ***escalated*** to the project-level roles for guidance.

Empowerment allows for rapid decision-making at the detailed level and thus rapid progress within a Timebox. Management by exception bridges the boundaries of that empowerment and ensures that, as and when the need arises, project-level roles are involved in making decisions which have a wider impact.

16.5 Planning throughout the Lifecycle

The following diagram describes at a very high level the focus of planning activity in each phase of the project. This planning activity is described in more detail in the following paragraphs



16b - Iterative development of plans

16.5.1 Planning Pre-Project

Planning Pre-Project is carried out at the programme/portfolio level and is focussed on when the Feasibility for the project will be assessed - based on the individual merits of the proposed project – and ensuring that the resources required to carry out the Feasibility assessment are available to do the required work.

16.5.2 Planning during the Feasibility phase

During Feasibility, a high-level investigation is carried out. Typically, at this point, there are a small number of requirements (fewer than ten), the solution is only an outline and there is still a lot to be discovered about the project. However, even with this level of information, it is both possible and sensible to plan in detail for the next phase, the Foundations. It is also possible to provide an approximation of the size and duration of the overall project, based on what is known at this point, but this can only be an educated guess. At this point, the Delivery Plan will describe the next few weeks (the Foundations) in detail, provide a very high-level outline for the first Increment and perhaps list the proposed dates for deployment of later Increments.

The detailed plan for the Foundations phase will include the timescale, the deliverables, the resources and the facilities needed. It is very useful to record some detail of Facilitated Workshops, (dates, participants, etc.). There is not sufficient information available yet to make detailed planning possible for the Evolutionary Development and Deployment phases, so there will be no detail about the number, duration and focus of the Timeboxes in the first Project Increment. Instead, this first cut of the Delivery Plan will probably simply state that the Evolutionary Development and Deployment phases will be of the order of x-y weeks/months, with a likely resource profile of z (Solution Developers, Business Ambassadors, Business Analysts and Solution Testers). At this point, given the level of information available, a significant margin of uncertainty is to be expected.

The business may have key dates in mind that reflect strategic business plans, so the Delivery Plan should also include an outline of what each of the proposed Project Increments are expected to achieve and any hard deadlines for them. It may not be possible to meet these deadlines and given the very limited information known about the requirements and proposed solution for the project it is not realistic to make any sort of commitment to delivery dates until the investigation during the Foundations phase is complete.

The answers to questions in the Project Approach Questionnaire (PAQ see Appendix B) will have an impact on how the project will be managed. Any special approaches to the project including any tailoring of the DSDM approach arising from the PAQ assessment are included in early drafts of the Management Approach Definition or Development Approach Definition.

16.5.3 Planning during the Foundations phase

During Foundations, the team carry out investigation to the next level of detail. By the end of Foundations, understanding of the requirements is a lot clearer. Each of the very high-level requirements from the Feasibility phase will have been expanded into more detail, so that typically requirements now number in tens and the overall MoSCoW priorities can be agreed. Since more information is now known about the requirements, the uncertainty is reduced and the accuracy of the estimate of work increases. However a range may still be provided, unless a fixed price estimate is required. In addition, more information about the business and technical background has been clarified. The foundation for the project is now a lot better understood and therefore it is now possible to create a version of the Delivery Plan with committed dates.

In the Foundations phase, planning focuses on three areas:

- Creating a schedule of Timeboxes for the first Project Increment along with resources required
- Defining the approaches to be used across the project for developing, and controlling the development of the solution
- Agreeing a strategy for deployment

The Delivery Plan is baselined at the end of the Foundations phase and the delivery date for at least the first Increment is committed.

As well as describing the number and likely duration of the Timeboxes, at least for the first Project Increment, the Delivery Plan also provides information on the probable focus for each Timebox, as well as the resources required to evolve the solution. The Delivery Plan does not provide the low-level detail of objectives and tasks for individual Timeboxes - that comes later, on a Timebox-by-Timebox basis as each Timebox is reached.

The detailed plans for Deployment of the solution are left until later in each Increment but the strategy for deployment and the high-level impact of, for example, business organisation change and training in the use of the new solution needs to be assessed as early as possible.

16.5.4 Planning Timeboxes as part of the Evolutionary Development phase

Timebox planning is carried out at the beginning of each Timebox and represents the lowest level of planning within a DSDM project. The Solution Development Team members are responsible for Timebox planning with plans being based on the objectives and outcomes agreed at the kick-off of each Timebox. Timebox Plans are based on task-level estimates that emerge as a result of detailed investigation of requirements and the decisions made on how these should be fulfilled. The plan itself is typically captured on a Team Board (or perhaps electronically where team members are not co-located) and will indicate who is responsible for doing what work to achieve the objectives of the timebox and generate the agreed outcomes.

The Team Leader is responsible for ensuring that all the work is covered by the plan and that resources are sufficient to do the majority of the work agreed. The Team Leader is also responsible for bringing to the attention of the Project Manager any significant issues that may result as the Timebox progresses especially if these impact on the agreed outcomes. Timebox duration is fixed at the kick-off – before the detailed planning is carried out. Therefore it is important that the requirements to be addressed are appropriately prioritised. This ensures that there is sufficient effort associated with the Could Have requirements (for the Timebox) to allow this work to be deferred if necessary, to ensure a coherent Solution Increment is delivered by agreed and immovable end date.

16.5.5 Planning for Deployment

As the detail of the solution emerges during the Evolutionary Development phase of the project, the plans for deployment of the solution can be considered in more detail. Deployment involves everything needed to transition the solution (or partial solution) into live operational use. The scope of the deployment activity varies considerably depending on complexity of the solution being deployed and the process used to deploy it. The Delivery Plan is updated with these deployment activities as they become clear. Care needs to be taken to ensure that plans for deployment are agreed far enough in advance to book necessary resources. These may include anything from the scheduling of rooms and individuals for training to ensuring access to a computer room on the 'go live' night and should always include roll-back options where appropriate in case of significant unexpected problems.

16.5.6 Planning benefits realisation activity, Post-Project

As the plans for the deployment of the solution become clear, the activities needed Post-Project to measure the benefits the solution will provide can also be planned. As the project will have closed down by this point, measurement of benefits is an activity that is owned by the Business Visionary as it is the Business Visionary who has responsibility for *owning the wider implications of any business change from an organisational and business process perspective and promoting the translation of the business vision into working practice*. In reality, a Business Analyst associated with the project can be involved with the planning of this activity whilst the project is still running and may also be involved with the actual measurement after it has closed down.

16.5.7 Incremental planning - re-visiting the Foundations phase between Increments

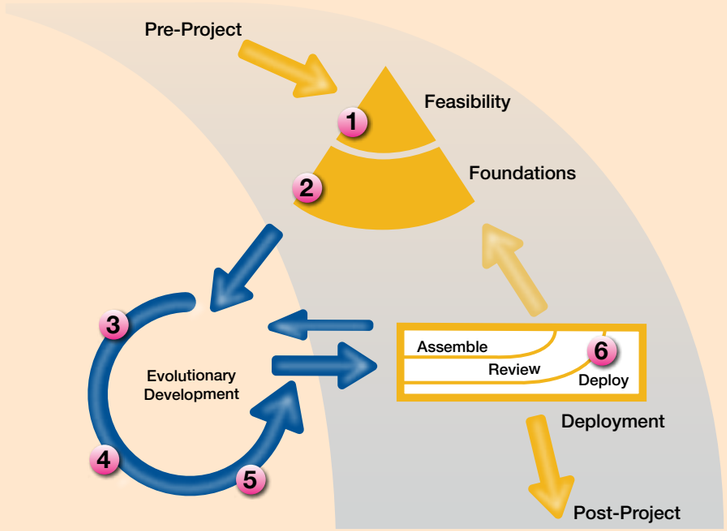
The incremental nature of the project means that it is often sensible to revisit the Foundations phase at the end of each Project Increment, following the back-arrow (A on diagram 16b) within the DSDM process to:

- Check that the project as a whole remains viable and should therefore proceed to the next Increment (based on validated actual effort and an updated, more informed view of the Business Case)
- Firm up plans for that Increment

The latter will involve a check on the validity and priority of requirements for the upcoming Increment and the scheduling the Timeboxes (as described above in Planning during the Foundations phase). It may also involve quickly revisiting the Management Approach (including a review of roles and responsibilities) and Development Approach to check that these are still appropriate.

16.6 Planning and Quality

To ensure successful delivery, quality is considered throughout the DSDM process.



16c - Quality control and assurance in the DSDM process

16.6.1 High level risk analysis (Feasibility) 1

Given that the Feasibility phase is focussed on understanding the likely costs and chance of success associated with a project it is essential to consider risks at a high level. A project that involves a high level of risk (either technical or business risk) will require a proportionate level of mitigation through reviews and testing. Also, when outlining the proposed solution to the project, it should identify any special qualities that the solution needs to have (performance, security, resilience and so on) which will in turn identify the need for specific types of non-functional testing. These factors (level of risk mitigation and the need for specialist tests) are a good indicator to the likely effort and resources that will be needed to assure the quality of our delivery.

16.6.2 Planning and high level analysis (Foundations) 2

During the Foundations phase, it is as important to include the quality perspective as it is to include individual opinions on how the solution will be constructed and what business activity it needs to support. This will derive benefits in two key ways:

- The quality viewpoint asks questions to assist in refining good high-level requirements from any requirements which may initially be too vague
- Three-way collaboration between Business roles (focussed on the need), the Solution Developers (focussed on design and build) and Solution Tester (focussed on proving the quality) will lead not only to better definition of requirements but also a shared understanding of these and shared ideas on how the solution is likely to evolve to demonstrably meet the business need

As an iterative part of carrying out the high-level planning during Foundations, elements of the Solution Foundations will evolve to help decision and agreement as to what aspects of the solution will need the most attention from a quality perspective. It also allows definition and agreement as to what non-functional testing will be needed. This is doubly beneficial, as it encourages the team to consider how those qualities will be built into the solution and verified on an on-going basis throughout the project. In poorly run projects, non-functional quality criteria are only defined and bolted-on at the end: whether or not the solution meets that standard is often pure luck and reworking can be astronomically expensive!

16.6.3 Detailed analysis and planning 3

Once the project moves into Evolutionary Development, the Solution Development Team will need to carry out more detailed analysis of the requirements and acceptance criteria that are being worked on, always in the context of what has been built so far. This involves looking at each of the acceptance criteria from different perspectives and with reference to different qualities before starting to build any given feature and to start preparing necessary quality assurance activity. It is valuable to capture what is decided in a light-weight fashion as we need to build up a picture of what has been delivered as the Evolving Solution progresses through multiple cycles of development.

16.6.4 Prepare and run 4

There is value in defining tests upfront and sharing those tests collaboratively with whoever is building the feature to be tested. Ideally, tests should form part of the detailed requirement. Carrying out test preparation also allows the Solution Development Team to identify the highest value tests so they can better prioritise testing activity.

Whilst running the tests, it is essential to capture the context, what was done and what was seen in a lightweight and fit-for-purpose way. In the IT environment, there are a growing number of tools which can help with this in an efficient and effective way. Testing only has meaning if the actions and the results can be demonstrated to a third party later on.

16.6.5 Assess quality and impact 5

As the tests are run, problems will be found. This is normal and to be expected. Finding big problems early on might cause the person carrying out the test to cut short the execution and instead to get together with the person who built the item and work out what needs to change. If there are no show-stopping problems, the person testing will use the time allocated as effectively as possible to get a good view on the quality of the deliverable. In a perfect world, this would be a binary, pass-or-fail exercise but in reality a more pragmatic, collaborative approach is often required in order to agree whether or not the item is fit for purpose and so can be considered 'done'.

At the end of the quality assurance activity the team need to do an impact assessment on what has been found. If there are residual problems then these need to be documented; there may be a need to change other parts of the design to accommodate this, or the team may decide to come back later to fix them or the cumulative effect may mean that the business case for the project is at risk.

16.6.6 Final end-to-end testing and assuring implementation 6

As much quality assurance activity as possible is built into the iterative development process. However, preparing to deploy a new system or component, it is important to test both the full package to be delivered and the process by which it will be delivered.

End-to-end testing may be the only time it is possible to fully test certain non-functional qualities of a solution (resilience, for example, typically needs a full solution in place before it can be tested). Naturally, it requires an appropriate place to test this, be that a proving ground or a test environment. Specialist tools or resources may also be needed to carry out what needs to be done.

From a technical perspective, every step in the process of deployment also needs to be assured. Walkthroughs of the process (a form of review) are beneficial to spot any errors or omissions. Once the process has been proven on paper, dress-rehearsals of the deployment (a form of testing) should be carried out. There also needs to be a way of verifying successful deployment once it has happened (which needs to be carefully designed so that it does not impact live operation) and a way to test back-out procedure in case that verification fails as a result of a significant problem occurring during deployment.

16.7 Summary

There have been concerns raised around some of the very informal styles of Agile, citing a perceived lack of planning and control. DSDM addresses these concerns, using as its base the wide experience of DSDM practitioners and DSDM in use within organisations of varying formality. This has allowed DSDM to evolve a robust but flexible framework to support planning and control within complex project, programme and corporate environments, but which can also work equally effectively on small simple projects. DSDM demonstrates that "Agile and control" and "Agile and planning" are concepts that work very effectively together, provided that Planning and Control are used based on an Agile mindset and Agile thinking.

17. Tailoring the DSDM Approach

17.1 Introduction

This chapter is drawn from the experiences of practising DSDM consultants. It describes some typical ways that DSDM has been adapted or tailored to meet circumstances that they have encountered. Hence each of the suggested actions is tried and tested but not necessarily to the same extent or with the same rigour as the rest of DSDM as described in this handbook.

17.2 The Project Approach Questionnaire (PAQ)

The Project Approach Questionnaire is used to identify areas where a project or its environment is not ideally suited to the DSDM approach. It can be used to negotiate changes to reduce risk and to improve the probability of success. Where changes cannot readily or quickly be made, or if too much change would be required to be accommodated at once, the PAQ can be a useful guide to the tailoring of DSDM to suit individual project needs. If, on collaborative completion of the PAQ, everybody either Strongly Agrees or Agrees with every statement, then the risk associated with using DSDM to manage the project is low. Tailoring is probably not required and DSDM as described in the first section of this handbook should work effectively.

If, however, there is disagreement with any of the statements, then there is probably some risk in trying to use DSDM straight out of the book.

In some cases, simple corrective action is all that is required to deal with the risk.

> *For example:*
>
> *statement 3 "The business vision is clearly stated and understood by all members of the project team". If the consensus is that the business vision is not clear and/or not understood, then arranging a session for the Business Visionary to share his/her vision and answer any questions mitigates the risk of evolving a solution not aligned with the business vision.*

In other cases the risk raised may not be as easy to resolve.

> *For example:*
>
> *statement 5 "The requirements can be prioritised and there is confidence that date and cost commitments can be met by flexing the scope of what is delivered." If the consensus is that a very high proportion (or all) of the requirements are genuinely "Must Have" – according to the MoSCoW rules – the DSDM approach will need significant adaptation to cope with something that contradicts a fundamental underpinning of the way it was designed to work.*

In the following section, each of the statements in the PAQ and their importance to the success of the approach are explained. Where appropriate, hints and tips towards resolving the issue of disagreement with that statement are provided.

There have been many examples of special configurations of DSDM, often created for blue-chip companies. Some of these are published as case studies. Ways of dealing with new challenges will be made available as part of the growing number of case studies available at www.dsdm.org

It should be noted that tailoring options may be interim solutions, to avoid imposing too much change for the team at one time. Thus, tailored elements may be subject to better alignment with DSDM at a later time.

One final point to note is that some risks or issues raised through the use of the DSDM PAQ may actually have a root cause that has nothing to do with the approach.

> *For example:*
>
> *statement 13 "The Solution Development Team members have the appropriate collective knowledge and skills to collaboratively evolve an optimal business solution". If the team doesn't have the skills required to build the solution, then regardless of what approach is chosen, the solution is highly unlikely to be built successfully.*

17.2.1 The Project Approach Questionnaire Statements

Statement 1

> *"All members of the project understand and accept the DSDM approach (Philosophy, Principles and Practices)"*

If the consensus response disagrees with this statement it is probably because some stakeholders have not been trained (if they are participants) or briefed (if they are less actively involved). It is important that the project team (Project-Level and Solution Development Team roles) fully understand the implications of this statement. Occasionally stakeholders who are fully informed about the DSDM approach still disagree with it, but this is extremely rare.

Suggested action
Organise training and briefings as required. Consider using DSDM Accredited Training Organisations or experienced DSDM practitioners certified to Advanced Practitioner level, or above, to assist where necessary.

Statement 2

> *"The Business Sponsor and the Business Visionary demonstrate clear and proactive ownership of the project"*

Senior business ownership of any project is essential. Without this, a project is likely to be starved of essential business resources (day-to-day jobs being deemed more important by default). This results in major issues, that can't be dealt with by the project team, remaining unresolved. Such unresolved issues either cause a project to stall or leave it exposed to high-risk assumptions or work-arounds. A committed Business Sponsor who really cares about the project is almost always willing and able to push for significant issues to be resolved where senior management action is needed.

The Business Visionary is responsible for making sure that all parts of the business impacted by the business vision understand and are bought in to the vision and remain aligned behind it. Managing business stakeholder expectation on a proactive and ongoing basis is key to the success of projects with a wide business impact.

Suggested action
There really is no work-around for a weakness in business ownership and vision. If all negotiation and coaching efforts to actively engage at least the Business Visionary fail (i.e if these critical roles simply refuse to engage in *their* project), the only technique to be applied here is necessarily harsh, and that is to refuse to start work on that project until the issue is addressed and, instead, to work on a project that does attract the right business ownership and commitment to success. Commercial and political considerations are likely to dictate whether this is even a sensible stand to make. However, for projects where the organisation responsible for building the solution and the sponsoring organisation belong to the same company, the argument of "the greater good of the company as a whole" may carry some weight.

With regard to managing business stakeholder expectation, a strategy for communication that the Business Visionary believes will meet the need, and is happy to play an active part in, should be considered as part of the Management Approach agreed during Foundations and enacted as the project proceeds. Stakeholders should also be invited to attend demonstrations of the Solution Increments at the end of each Timebox where this helps manage their expectations. (see Statement 16)

Statement 3

> *"The business vision driving the project is clearly stated and understood by all members of the project team"*

All day-to-day project decisions at all levels in the project should be checked against the business vision. Even if the decision seems small or insignificant simply considering "Does this help move us closer to achieving the business vision?" often avoids wasting precious time and effort. If the answer is "Yes, it helps" then the decision

is valid, if it is "no" or "not sure" then a follow up question of "So why are we doing this here and now?" should be asked. For anything likely to take up more than a couple of hours it is always advisable to quickly consult an appropriate business or technical authority. Without a clear statement and common understanding of the business vision the thought/questioning process cannot happen and there is serious risk that time and effort will be wasted working on things with limited or peripheral value.

Suggested action
Speak to the Business Visionary, or arrange a session for him/her to share their vision and answer any questions. This should be all that is needed to set the scene for the thought and decision-making described above. It is an explicit responsibility of the Business Visionary to "communicate and promote the business vision to all interested and/or impacted parties" and to ensure that the project remains aligned to it by "monitoring progress of the project in line with the business vision". Consider creating a simple poster to place above the Team Board to help keep the vision visible to all.

Statement 4

"All project participants understand and accept that on-time delivery of an acceptable solution is the primary measure of success for the project"

There are two key considerations associated with this statement.

The first consideration relates to the business driver behind the project and the importance from a business perspective of on-time delivery. Most businesses under most circumstances want to understand at the outset what a project is going to cost and how long it is going to take. These are very reasonable demands as the cost and the date that benefits will start to be realised have a direct impact on the Business Case for the project. Over-running budget and, particularly, time, to any significant degree could seriously damage the Business Case or even the business as a whole if a critical deadline is missed.

The second consideration relates to the control of the project. Even if the first consideration (immediately above) is not important, working to a genuinely fixed end date for a project, or at least for a Project Increment, provides an anchor for the combined practices of MoSCoW prioritisation and timeboxing which forms the primary mechanism of control over timely delivery of the solution. It also keeps the Iterative Development practice properly focussed on the business need by discouraging a quest for perfection through iterations with progressively diminishing business value.

Suggested action
Even if there is no hard business deadline for delivery of the solution, it is usually better to plan and resource a project on the assumption that there is. Without a delivery target that everybody buys into, the project risks losing focus and running out of control.

Statement 5

"The requirements can be prioritised and there is confidence that cost and time commitments can be met by flexing the scope of what is delivered"

This statement is closely related to Statement 4, as it is the ability to flex the scope of what is delivered that allows a DSDM project to guarantee an on-time delivery.

Every effort should be made to follow the MoSCoW rules for prioritisation of requirements but it is worth remembering that even if the top-level requirements (that make up the Prioritised Requirements List baselined in the Foundations phase of the project) suggest the rules are being broken, it is possible that flexibility exists in the detail of those requirements.

For example:

a Must Have requirement to "manage an appointment diary" is likely to break down into sub-requirements to "make", "change" and "cancel" appointments. At this lower level it likely that the ability to make and cancel appointments would be Must Have whereas the abilityto change an appointment would have a lower priority as there is an obvious work-around (cancelling the original appointment and making a new one).

Suggested action
If efforts to find sufficient contingency in the scope of the requirements fail, consider adding more "traditional"-style contingency based around time and cost. In practical terms, this would involve:

- Creating one or more 'contingency Timeboxes' that are added to the end of the project
  - The *committed* timescale for the project includes the contingency but the Delivery Plan should clearly reflect an earlier *target* delivery date that does not include the contingency
- Managing the project in the normal way once development starts. However, instead of de-scoping the least important requirement to protect the *target* delivery date if that becomes necessary, instead the requirement is pushed out to the first available contingency Timebox

If using this tailoring suggestion, everybody's focus must remain on the *target* end date and this target end date must be treated as if it were the real delivery date. Without this focus, the attitude of "It doesn't matter too much because we can always put it in the contingency Timebox" risks the contingency being used up too quickly and carelessly.

Statement 6

"All members of the project team accept that requirements should only be defined at a high level in the early phases of the project and that detail will be emerge as development progresses"

Defining detail too early in a project causes more problems than it is intended to solve. All Agile approaches exploit the concept of emerging detail to allow the best solution to evolve. Discussion of detail of a high-level requirement may help cement an understanding of that requirement and help estimate the effort to fulfil it. However, it will also provide a false sense of security as the majority of change to requirements in a project happens in the detail. The reality is that detail defined too far in advance risks being inaccurate when the work on the detail is due to start. This inaccuracy may be as a result of a subtle shift in business need. It may be due to a deepening understanding of what is, or is not possible, based on what has happened up to this point. It may be due to earlier assumptions proving to be untrue. For this reason, DSDM advocates deferring detailed investigation and detailed decisions to the last responsible moment.

Suggested action
Hold early discussions at whatever level of detail is needed to help drive out a shared understanding but do not capture that detail. Have the discussion again closer to the time. Subsequent discussions will probably be shorter as people refresh their memories of what was discussed previously. But these also offer an opportunity to change that detail without being constrained by what was previously assumed or wasting time having to formally change what was previously formally, and pointlessly, defined and agreed.

Statement 7

"All members of the project team accept that change in requirements is inevitable and that it is only by embracing change that the right solution will be delivered"

The DSDM philosophy that *best business value emerges when projects are aligned to clear business goals, deliver frequently and involve the collaboration of motivated and empowered people* underpins the validity and importance

of this statement. It is important to ensure that the project remains focussed on the fundamental need and business vision that justified the investment in it. However beyond that, the Project Team needs to embrace any change needed to deliver optimum business value within constraints of fixed timescales and cost. "What we thought we were going to do" is irrelevant when considered in the light of "what we need to do, now, to build a valuable solution".

Suggested action
If commercial arrangements dictate that a 'fixed price for a fixed specification' model should be applied to the project rather than a more collaborative approach, it is important to recognise that at the working level this will not be a DSDM project. Under such circumstances all that can be done is to segment the project into small deliverable chunks and to ensure the supplier is focussed on and is paid to deliver only what is specified in each chunk. The specification of detail for any given chunk should be left to the last responsible moment and should be informed by what has already been delivered together with the very latest thinking on what is needed. The later chunks in the project should reflect the least valuable features of the product being built by the supplier. Arrangements with the supplier(s) should allow for any changes that may be needed to the product they have built 'to specification' in an early chunk to be traded off against later work. This will, at least, force a change-tolerant incremental approach that will help mitigate the risk of losing control of timescales and/or costs.

Statement 8

"The Business Sponsor and Business Visionary understand that active business involvement is essential and have the willingness and authority to commit appropriate business resources to the project"

In a DSDM project there is no detailed specification upfront (compared with a traditional project approach where a detailed specification is created by contributions from many business representatives). Without detailed guidance during development, those developing the solution can only guess at the detail of what is needed which would result in a significant risk of delivering a solution that does not meet the business need. In DSDM, active business involvement means that business roles (Business Ambassadors and Business Advisors) must be involved throughout the project, often on a day-to-day basis, sufficient to:

- Provide detailed guidance on the meaning of requirements
- Understand team plans for the Evolving Solution
- Provide feedback on each step towards delivering a fit-for-purpose solution, acknowledging what is right and explaining what is not

Involvement of Business Ambassadors and Advisors at the time detail is needed cannot be negotiated away, but there is room to negotiate on how much involvement might be needed, and the frequency and form that might take is discussed under the next statement. (Statement 9)

Suggested action
The first and best course of action is to try to secure the necessary Business Ambassador and Advisor time. It is helpful to agree up front the amount of time and the level of commitment expected, at least for the Business Ambassador. This helps inform the business what level of commitment is expected and tests commitment.

One example: During Evolutionary Development, Business Ambassador to:

- *Attend Daily Stand-ups (15 minutes) every day whenever possible.*
- *Be available at 9.30 each morning on phone to answer questions (maximum 20 minutes).*
- *Be available 2-3 days every 2 weeks to attend (in person or on webinar) Timebox events (Kick off, Close out, Reviews)*

It is important to stress that projects cost money and the single biggest cost is usually paying for the time of the people engaged in building the solution. Business resources are critical to the success of the project and should be budgeted for in exactly the same way as the other members of the Solution Development Team. This means that the Business Sponsor may need to pay for somebody to carry out some of the day-to-day responsibilities of a Business Ambassador in order to give them the time they need to spend on this project. It is normally the case that the best person to take on the Business Ambassador role – the main day-to-day business decision-maker on the project - is also often someone who is very valuable to the business area concerned. This means that negotiating commitment of their time needs careful planning and advanced preparation. The answer often lies in other members of the business area picking up key responsibilities whilst handing off responsibilities most easily delegated to somebody more junior. Some businesses choose to delegate simple but time-consuming tasks to a temporary staff member, hired only for the duration of the project but treated as part of the project cost.

If it is genuinely not possible to allocate business resources to work collaboratively with the rest of the Solution Development Team on an on-going basis, then it may be worth considering the approach described in the "Specification-led DSDM projects" tailoring white paper available at www.dsdm.org

Statement 9

"It is possible for business and solution development members of the Solution Development Team to work collaboratively throughout the project"

Statement 8 dealt with business roles being allocated and available to the project as needed to guide the detailed development of the solution. This statement addresses the issue of those roles, and the rest of the Solution Development Team, being able to work in a collaborative way.

With regards to business engagement in the Iterative Development process, it would be ideal if the business roles were ready, willing and able to engage in face-to-face conversation with the Solution Development roles immediately and whenever their guidance is needed. This 'instant access' would optimise the efficiency and effectiveness of the Iterative Development practice, by minimising the risk of evolving the solution in the wrong way and having to re-work it to make it right. However, such access is rarely achieved and in reality may not represent effective use of business resource time, as it implies they must be co-located with the rest of the team and do little more than sit and wait to be engaged in development or testing activity. There is also a risk that by being removed from their business colleagues, the Ambassador may lose touch with day-to-day happenings in their business area. Although some exceptional projects do require a full-time Business Ambassador commitment, the majority of projects require active involvement of no more than half of an average day as a maximum. The amount of Business Ambassador time should, therefore, be agreed on a project-by-project basis.

What is actually needed is reasonable access throughout the day. This may be perhaps face-to-face (or by telephone or on video conference) at the Daily Stand-up and for a short period afterwards with availability on the telephone for the rest of the day. On occasional days in a month, more intense collaborative activity may be needed, for example in workshops to discuss the detail of requirements towards the beginning of a Timebox or to carry out end-to-end business acceptance testing (allowing for iterative issue resolution) towards the end of a Timebox.

With regard to collaborative working, more generally, it is important that all roles are able to work collaboratively on the Evolving Solution. It is important to ensure that the working environment can support this – ideally by allowing team members to be co-located in the workplace and in less ideal circumstances by providing technology to simulate this.

Suggested action
If access to business resources presents a challenge in terms of time, e.g. where a Business Ambassador is available for a limited number of hours a week, try to formalise the structure of the Timebox so that intense engagement during the Investigation and perhaps Consolidation steps can be planned. Also agree a short period in each day, ideally around the Daily Stand-up, when Solution Developers and Testers can interact with the Business Ambassador.

If access to business resources presents a challenge in terms of geography, e.g. where business resources are in a different building, town, country or even continent, then technology to assist in collaboration will be needed. Video and teleconference facilities are an obvious place to start but other tools are available to help with collaborative working, such as collaborative modelling tools or a virtual Team Board that all can see and interact with.

In extreme, but not uncommon, circumstances – for example when working across time-zones more than 4 hours apart, special tailoring of roles and responsibilities to deal with communication may be required. The case study on "DSDM projects with off-shore development" is available as a tailoring white paper from www.dsdm.org.

Statement 10

"Empowerment of all members of the Solution Development Team is appropriate and sufficient to support the day-to-day decision-making needed to rapidly evolve the solution in short, focussed Timeboxes"

A framework of empowerment underpins the DSDM way of working. It is important that members of the Solution Development Team have the knowledge and experience necessary to make day-to-day decisions about how the Solution should evolve and that they are empowered to do so. If members of the team have to keep referring out to 'higher authorities' to make or ratify such decisions, the efficiency of the Iterative Development practice, the effectiveness of timeboxing will be seriously compromised and the promise to *Deliver on Time* will be put at risk.

Suggested action
There is no effective work-around for disagreement on this statement. A way must be found to establish a framework of empowerment even if this takes some time to achieve. The 'higher authorities' will need to engage more actively in the project in the first instance with the intention of gradually handing decision-making power to the team as they gain competence and confidence to assume that responsibility.

Statement 11

"The DSDM roles and responsibilities are appropriately allocated and all role holders understand and accept the responsibilities associated with their role"

DSDM has carefully defined roles and responsibilities. One person may hold more than one role and one role may be shared between more than one person. It is useful to use DSDM role names and descriptions, particularly with business roles, as the title re-enforces their main responsibility. When allocating roles it is important to review the responsibilities associated with that role when determining who the role holder should be, in order to get the best fit.

Suggested action
Formal training in DSDM is recommended for all roles. Those people actively working on the project on a day-to-day basis, typically the Project Manager and all members of the Solution Development Team, should attend a DSDM Practitioner course. Roles less actively engaged should attend a DSDM Foundations course. Both training courses explain to each individual the responsibilities of their role in the context of the other roles and the DSDM approach to the project. It is also important to ensure the person in the role understands and is comfortable with what is expected of them on a project-by-project basis.

In some cases, it may be sensible to transfer one or more responsibilities from one role to another to help achieve a good fit of responsibilities to a given individual.

> *For example:*
>
> *if the best person to fulfil the majority of the Business Visionary responsibilities is more junior in the organisation than would normally be expected, it may be appropriate to transfer responsibilities such as "owning the wider implications of any business change from an organisational and business process perspective" and "ensuring business resources are available to the project as needed" to the Business Sponsor.*

Under certain circumstances it may be appropriate for one person to hold more than one role. For example, in a smaller project it might be appropriate for the Business Visionary and Business Ambassador roles to be held by the same person. Sometimes, the person holding the Technical Coordinator role will play a more 'hands on' role in developing the solution – in which case the individual concerned may also be a Solution Developer.

A DSDM coach – somebody with real practical experience of using DSDM in a variety of circumstances – can help with roles assignment and tailoring and can also help individuals understand and properly fulfil their roles

Statement 12

"The Solution Development Team has the appropriate collective knowledge and skills (soft skills and technical skills) to collaboratively evolve an optimal business solution"

DSDM works best in circumstances where all team members are experienced and empowered to shape the solution and where they have the necessary soft skills to communicate and negotiate effectively with their teammates. Ideally, solution roles will be technically multi-skilled – willing and able to work across all solution development disciplines (analysis, design, build and test). A key characteristic of an effective collaborative team stems from the ability and willingness of team members to support each other.

Suggested action
With regard to technical skills, at the project level there are no general work-around options to suggest, except to "make the best of what you have". Where work can only be done by a single individual, try to ensure it is broken down to a level where something tangible can comfortably be delivered in a Timebox. For example, avoid putting a single 10 man-day task into a 10 working day Timebox.

At the organisation level, if DSDM (or any other Agile approach) is being adopted as the default way of working on projects, effort should be made to invest in people; and through training and time for personal growth, encourage project workers to become more multi-skilled over time. The value of training and coaching soft skills where these are weak also represents an excellent investment in people because, as well as being essential in an Agile project team context, these skills can be valuably applied much more widely.

Statement 13

"Solution Development Team members are allocated to the project at an appropriate and consistent level sufficient to fully support the DSDM timeboxing practice"

By default, DSDM assumes that the Solution Developers and Testers are allocated full time to a project at least for the duration of a Project Increment. Where individuals are not full time on a project it is assumed that the work of the project is always their top priority. Where the Solution Development Team is made up of part-time individuals for whom other work takes priority, it is very difficult to make the timeboxing practice work.

Suggested action
Where resources are only available part-time, first try to secure formal agreements as to how many hours per day, per week or per Timebox they will spend on the project. Ideally agree specific times, e.g. 9am to 12:30pm Monday through to Thursday. Where multiple team members are part-time, try to synchronise agreements in order to allow team members to work collaboratively.

Agree objectives and schedule the work in Timeboxes to match availability of resources: be careful not to over-commit, especially in circumstances when availability may be unpredictable. In cases where resource availability is very unpredictable, do not commit to a delivery date, just agree to try and meet a target. If this is unacceptable to the Business Sponsor, make it their problem to negotiate a more appropriate resource profile; perhaps by using contract resources.

Try to make the work as granular as possible – the smaller each piece of work, the more likely it will get finished within the boundaries of the Timebox. Avoid the temptation to make the Timeboxes longer as this dilutes what little focus on delivery there is.

Statement 14

"Tools and collaborative working practices within the Solution Development Team are sufficient to allow effective Iterative Development of the solution"

Collaboration and empowerment underpin DSDM's Iterative Development and timeboxing practices.

Suggested action
Whenever appropriate, face-to-face communication should be encouraged at all levels. The supporting practices of Facilitated Workshops and Modelling play a significant part in making this effective.

Where teams are not co-located the suggestion offered for Statement 9 applies, i.e. using technology to assist in collaboration, and considering additional roles to focus on communication.

Statement 15

"All necessary review and testing activity is fully integrated within the Iterative Development practice"

Chapter 11 (Iterative Development) and Chapter 16 (Planning and Control) comprehensively cover the rationale for this statement. The essence of the guidance provided is to start thinking about how the solution will be tested as early in the lifecycle as possible and defining and executing a strategy for testing that gets as close as possible to delivering fully tested Solution Increments at the end of every Timebox.

Suggested action
Where fully integrated testing cannot be achieved – perhaps due to challenges of integrating the outputs of two or more Solution Development Teams, or where testing is carried out by a separate off-shore team – it may be sensible to set up a parallel stream, ideally made up of dedicated resources, focussed on testing and fixing any defects that may emerge as a result of that testing. Timeboxes in the testing stream would be offset from those in the main development stream so that the output of the development Timeboxes that have just finished would be the input for a testing Timebox that is just about to start. As much testing as possible should still be carried out by the original development teams and the Technical Coordinator will need to be actively engaged in ensuring that solution and test design across teams is compatible. This should lower the risk of significant rework emerging from defects discovered by the integration/testing team.

Statement 16

"Project progress is measured primarily through the incremental, demonstrable delivery of business value"

The primary output of each Timebox should be a demonstrable increment of the Evolving Solution. Every effort should be made to ensure that this is the case. If it is, then measurement of progress is easy and fully transparent with all interested stakeholders able to see tangible progress as a result of the work of the Timebox.

The demonstration of the Solution Increment at the end of a Timebox is an excellent way of keeping all stakeholders informed of progress and provides them with a real opportunity to understand in detail how they will be impacted by the solution once it has been deployed.

Suggested action
In most cases, if Solution Increments cannot be demonstrated, it is as a result of a poor strategy for Iterative Development or poor application of the timeboxing practice.

In the rare circumstances where it is genuinely not possible to deliver a demonstrable Solution Increment, think about what can be demonstrated. It is vitally important that all stakeholders have confidence that the project is moving towards a successful conclusion and is on track to deliver a valuable solution in the timeframe and for the budget agreed. Do whatever is necessary (within reason) to allow this to be achieved.

For example:

an early Timebox was proving the ability to communicate with a new customer located several hundred miles away, so that financial transactions could be transmitted in a subsequent Timebox. The early Timebox transmitted a simple "Hello" message which was then printed at the remote site, as proof that the communication channel was now in existence.

Consider making the demonstrations at the end of each Timebox open to anybody who is interested in what is happening in the project and, where appropriate, use it as a way of helping keep stakeholders on board.

Statement 17

"There are no mandatory standards or other constraints in force that prevent the application of the DSDM Philosophy and Practices on this project"

Many organisations assume that the "fixed price for a fixed specification" model is the best foundation for a commercial agreement as it appears to transfer the risks associated with the project to the supplier organisation. For many projects this transfer of risk is an illusion as all of the risks and issues associated with the traditional Waterfall way of working – which the DSDM approach was designed to address – are actually exaggerated by this commercial framework. Whilst it is true that there may be somebody to pursue for compensation when a project goes wrong, such litigation is extremely rare as 'blame' for failure can rarely be put exclusively on one party to the agreement. And the end result is still that the business "loses", since it has lost time and still does not have a viable business solution to the problem.

Suggested action
Consider the DSDM Principles, Practices, Roles and Responsibilities. If any of these are seriously undermined by commercial agreements, then the DSDM approach may not be the best approach to use for the project. Whilst the 'fixed price for a fixed specification' model for project is flawed, it is well understood, and trying to force DSDM to work under the restrictions imposed by it are likely to make the project even more risky than it would otherwise be. For many organisations, the procurement team are only set up to deal with a traditional style contract, and simply do not have the mindset or the willingness to take on the risk associated with a different style of contractual relationship, even where it should directly benefit the organisation.

At time of first printing of this handbook in mid-2014, work has been going on for some time to evolve an effective Agile contract framework but this is a work in progress. Check www.dsdm.org for resources and links to other sources for contractual models available that suit your project.

17.3 Summary

DSDM is a flexible framework for building and delivering business solutions. The advice above describes ways of tailoring DSDM to overcome issues where the project or the environment in which it exists are not ideally suited to the Agile (empowered, collaborative, iterative, incremental) way of working. In the majority of cases where issues are identified, what is really needed is a change of mind-set of those involved rather than a customisation of the project approach. The full value that can be gained from using DSDM will not be achieved if too many compromises are made, so every effort to get buy-in to the approach, and all that that entails in terms of working practice, should be exhausted before starting to adapt it.

Appendix A
Glossary

Term	Abbreviation	Detail
80:20 rule		A rule of thumb stating that 80% of consequences stem from 20% of causes. Also known as the Pareto Principle; it advocates pragmatism on a DSDM project. The value of the Pareto Principle is that it reminds you to focus on the 20% that matters.
Agile		A style of working where requirements and solutions evolve through collaboration between self-organising, cross-functional teams. Agile promotes adaptive planning, evolutionary development and delivery, a timeboxed, iterative approach and encourages rapid and flexible response to change.
Agile Manifesto		The Agile Manifesto defines the approach and style that is fundamental to all Agile approaches. It was created in 2001, at a summit attended by representatives of all the Agile methodologies.
Benefits Assessment		A DSDM Product. It describes how the benefits have actually accrued, following a period of use in live operation.
Business Case		A DSDM Product. Baselined at the end of the Foundations phase, it provides a vision and a justification for the project from a business perspective.
Bottom Up		A style of estimating. Using this approach, each component is estimated individually and then the estimates are summed to find the total effort.
Cycle		DSDM defines Iterative Development as an informal cycle of "Thought, Action, Conversation".
Delivery Plan		A DSDM Product. It provides a high-level schedule of Increments for the project and, at least for the first/imminent Increment, the timeboxes that make up that Increment.
Development Approach Definition	DAD	A DSDM Product. Baselined at the end of the Foundations phase, it provides a definition of the tools, techniques, customs, practices and standards that will be applied to the Evolutionary Development of the solution.
Deployed Solution		This is a baseline of the Evolving Solution, which is deployed into live use at the end of each Project Increment.
Deployment		The DSDM lifecycle phase which focuses on getting the solution (or part of it) into operational use.
Development Timebox		A fixed period of time, part of Evolutionary Development, where development and testing of the Evolving Solution takes place. Typically 2-4 weeks long. See Timebox.
Done		A common term used in Scrum - an item is "Done" (completed) when it meets all the criteria that have been defined for it ("Definition of Done".) Done is binary - an item is either Done or Not Done.
Evolutionary Development		The DSDM lifecycle phase used iteratively and incrementally to investigate the detailed requirements and evolve them into a viable solution

Term	Abbreviation	Detail
Evolving Solution		A DSDM Product. It is made up of all appropriate components of the final solution together with any intermediate deliverables necessary to explore the detail of requirements and the solution under construction. At any given time, such components may be either complete, a baseline of a partial solution, or a work in progress. They include, where valuable: models, prototypes, supporting materials and testing and review artefacts.
Feasibility		The DSDM lifecycle phase which gives the first opportunity for deciding whether or not the project is viable from a technical and/or business perspective.
Feasibility Assessment		A DSDM Product. It provides a snapshot of the evolving business, solution and management products as they exist at the end of the Feasibility phase. It may be expressed as a baselined collection of the products or as an executive summary covering the key aspects of each of them.
Fit for Purpose		Something that is good enough to do the job it was intended to do.
Foundations		The DSDM phase to establish firm and enduring foundations from the three perspectives on a project of business, solution and management.
Foundation Summary		A DSDM Product. It provides a snapshot of the evolving business, solution and management products as they exist at the end of the Foundations phase. It may be expressed as a baselined collection of the products or as an executive summary covering the key aspects of each of them.
Function / Feature		See Requirement.
Increment		An element of the Evolving Solution, comprising a collection of one or more features which, as a group, have meaning/value for the business. One or more Increments may form a Release.
Increment Timebox		Timeboxing can be applied at increment level and an Increment Timebox comprises the time fixed by the sum of the Development Timeboxes for this Increment. See Timebox.
Instrumental Success Factor	ISF	A key behaviour or style of working that is seen as instrumental to position DSDM projects for success . Where these factors cannot be met, they represent a significant risk to the DSDM approach.
Iteration		1. A general term for working in a cyclic way, where several attempts are made in order to get a more accurate or beneficial result. 2. One cycle of development and testing which takes place (one or more times) inside a Timebox and which finishes with a Review. 3. In eXtreme Programming, Iteration equates to a DSDM Timebox.
MoSCoW		A DSDM prioritisation technique, mainly used on requirements although also useful in other areas (such as testing). M stands for Must Have, S stands for Should Have, C stands for Could Have and W stands for Won't Have *This Time*.

Term	Abbreviation	Detail
Management Approach Definition	MAD	A DSDM Product. Baselined at the end of the Foundations phase, it reflects the approach to the management of the project as a whole and considers, from a management perspective, how the project will be organised and planned, how stakeholders will be engaged in the project and how progress will be demonstrated and, if necessary, reported.
Minimum Usable SubseT	MUST	The minimum set of requirements needed to deliver a usable solution – the "Worst Case" basic deliverable. The Minimum Usable SubseT is defined as the Must Haves. Provided the (MUST) MoSCoW rules are properly applied, delivery of the Minimum Usable SubseT is guaranteed.
Post-Project		The DSDM phase which takes place after the last planned Deployment. It is used to assess the business value delivered by the project.
Pre-Project		The DSDM phase where the initial idea or imperative is formalised in order to initiate a project.
Principle		A 'natural law' which acts as an attitude to take and a mindset to adopt on a DSDM project.
Prioritised Requirements List	PRL	A DSDM Product. Baselined at the end of the Foundations phase it describes the requirements that the project needs to address and indicates their priority with respect to meeting the objectives of the project and the needs of the business.
Project Approach Questionnaire	PAQ	The DSDM questionnaire, based on the Instrumental Success Factors which helps flag potential risks to a successful DSDM project.
Project Governance Authority		A panel of corporate decision-makers who decide whether projects should proceed or not.
Project Review Report		A DSDM Product. Updated at the end of each Increment it: captures the feedback to confirm what has been delivered and what has not; captures learning points from the retrospective focusing on the process, practices employed and contributing roles and responsibilities; where appropriate it describes the business benefits that should now accrue through the proper operation of the solution delivered by the project up to this point. After the final Project Increment a Project Retrospective, in part informed by these Increment reviews, is prepared as part of the closure of the project.
Project Timebox		Timeboxing can be applied at project-level and a Project Timebox comprises the time fixed by the sum of the Increment Timeboxes for the project. See Timebox.
Prototype		A piece of work that demonstrates how a given objective can be or has been achieved or to prove a concept.
Release		A collection of Features (developed and tested elements of the Evolving Solution) being deployed into operational use. A Release may comprise one or more Increments.
Requirement		Something the final solution needs to be able to do (functional requirement) or do to a certain level (non-functional requirement). Similar words: function, feature, User Story.

Term	Abbreviation	Detail
Retrospective		A Facilitated Workshop to look back on a recent event and to assess what went well and what could be improved.
Return on Investment	ROI	The concept of an investment of some resource which yields a benefit to the investor.
Solution Architecture Definition	SAD	A DSDM Product. Baselined at the end of the Foundations phase, it provides the design framework for the solution.
Scope		A description of what the solution will do and what it will not do. This could be a list of features and/or a description of areas of the business that may or may not be affected.
SCRUM		One of the Agile approaches, with a strong focus on the team management process. Scrum's focus is on a flexible, holistic product development strategy.
Servant-Leader		Servant-Leader is the style of leadership that Agile projects aspire to, in particular from the Project Manager and Team Leader roles. A servant-leader shares power, puts the needs of others first and helps people develop and perform as highly as possible.
Stakeholder		A person, group, organisation, member or system who either affects or is affected by actions taken by the Project or the Team.
Story		See User Story.
Test-Driven Development	TDD	An approach whereby a test is written before the solution is built, thus ensuring the requirement is understood and testable. TDD aims to encourage simple designs and inspire confidence. It is most commonly applied in an IT environment but is now gaining interest as a technique outside IT.
Team Board		A large graphical representation of project/Timebox information kept plainly in sight within an Agile team's shared workspace. It shows anyone who views it information they care about, and thus avoids the need to keep asking the team for information. This ensures more communication with fewer interruptions. Team Boards can contain most types of charts used in Agile development. Burn-down charts, task boards, planning boards and storyboards are among the possibilities. A Team Board is usually hand-drawn or printed but can also include computer-generated charts and electronic displays. (Sometimes called Information Radiator, Big Visible Chart or KanBan Board).
Terms of Reference	ToR	A DSDM product created Pre-project. It is a high-level definition of the over-arching business driver for, and top-level objectives of, the project.
Timebox		A fixed period of time, at the end of which an objective has been met. The objective would typically be a deliverable of some sort. Typically Timeboxes operate at development level, but timeboxing can also be applied at project and increment level. A timebox is managed by adding or removing content in order to meet the timebox objective and the deadline. (See also Sprint [Scrum] and Iteration [XP]).

Term	Abbreviation	Detail
Timebox Plan		A DSDM Product. It is created for each Timebox. It elaborates on the objectives provided for that Timebox and details the expected deliverables, along with the activities to produce those deliverables and the resources to do the work. The Timebox Plan is created by the Solution Development Team and is often represented on a Team Board as work to do, in progress, and done.
Timebox Review Record		A DSDM Product. It is created for each Development Timebox, capturing the feedback from each review that takes place during that Timebox. It describes what has been achieved up to that point together with any feedback that may influence plans moving forwards. Where appropriate, e.g. in a regulated environment, it may provide a formal auditable record of review comments from expert Business Advisors and other roles.
Top-Down		A style of estimating using approximate sizings and groupings. For example, estimating 10 small components at typically one day each, 20 medium components at typically three days each, three complex components at typically five days each. These groups are summed to give an approximate estimate for a solution where the low level detail is probably still unknown.
Transparency		This describes openness, communication, and visibility. Transparency means operating in such a way that it is easy for others to see what actions are being performed and what progress is being made.
User Story		A requirement expressed from a user point of view and with associated acceptance criteria. The usual format is: As a <role> I want <requirement / Feature> so that <benefit to be gained>.
eXtreme Programming	XP	One of the Agile approaches with a strong focus on technical (IT) development techniques. XP is intended to improve software quality and responsiveness to changing customer requirements.

Appendix B
Project Approach Questionnaire (PAQ)

DSDM Project Approach Questionnaire (PAQ)		Collective opinion				
Ref	Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	All members of the project understand and accept the DSDM approach (Philosophy, Principles and Practices)					
2	The Business Sponsor and the Business Visionary demonstrate clear and proactive ownership of the project					
3	The business vision driving the project is clearly stated and understood by all members of the project team					
4	All project participants understand and accept that on-time delivery of an acceptable solution is the primary measure of success for the project					
5	The requirements can be prioritised and there is confidence that cost and time commitments can be met by flexing the scope of what is delivered					
6	All members of the project team accept that requirements should only be defined at a high level in the early phases of the project and that detail will be emerge as development progresses					
7	All members of the project team accept that change in requirements is inevitable and that it is only by embracing change that the right solution will be delivered					
8	The Business Sponsor and Business Visionary understand that active business involvement is essential and have the willingness and authority to commit appropriate business resources to the project					
9	It is possible for the business and solution development members of the Solution Development Team to work collaboratively throughout the project					
10	Empowerment of all members of the Solution Development Team is appropriate and sufficient to support the day-to-day decision-making needed to rapidly evolve the solution in short, focussed Timeboxes					
11	The DSDM roles and responsibilities are appropriately allocated and all role holders understand and accept the responsibilities associated with their role					
12	The Solution Development Team has the appropriate collective knowledge and skills (soft skills and technical skills) to collaboratively evolve an optimal business solution					
13	Solution Development Team members are allocated to the project at an appropriate and consistent level sufficient to fully support the DSDM timeboxing practice					
14	Tools and collaborative working practices within the Solution Development Team are sufficient to allow effective Iterative Development of the solution					
15	All necessary review and testing activity is fully integrated within the Iterative Development practice					
16	Project progress is measured primarily through the incremental, demonstrable delivery of business value					
17	There are no mandatory standards or other constraints in place that prevent the application of the DSDM Philosophy and Practices on this project					

Appendix C
Index

This index is in alphabetical, word by word order. It does not cover the Contents list or Foreword. Location references are to chapter and section number, e.g.

benefits realisation, 16.5.6

indicates that information on the realisation of benefits can be found in Chapter 16, section 16.5.6.

Abbreviations: App = Appendix; Fig = Figure; Tab = Table