

The DSDM Agile Project Framework for Scrum

Revised and Updated – June 2014



White Paper

Produced by:

**Andrew Craddock, Barbara Roberts,
Julia Godwin, Dorothy Tudor, Keith Richards**
for the DSDM Consortium

Contents

Introduction	2
Introducing the DSDM® Agile Project Framework (AgilePF)	2
Introducing DSDM	2
Introducing Scrum.....	3
AgilePF for Scrum	4
Philosophy	4
Agile Values	4
Principles.....	5
Variables	8
Lifecycle	9
Roles	11
Products (Artefacts)	17
DSDM Agile Project Framework Techniques	19
Introduction	19
Facilitated Workshops.....	19
Modelling.....	20
MoSCoW Prioritisation (optionally applied to the Product Backlog).....	20
DSDM Timeboxing (optionally applied to a Sprint)	20
In Summary.....	22
About the Authors.....	22

Introduction

Introducing the DSDM® Agile Project Framework (AgilePF)

The DSDM Agile Project Framework is the latest version of the DSDM method, designed in such a way as to allow easy integration with other Agile approaches such as Scrum and Agile software engineering practices such as those advocated by eXtreme Programming. It also retains a key strength from previous versions of the method designed to allow easy adoption by larger organisations where a full project approach, with formal governance, is either expected or will add value.

This pocket book describes a version of the DSDM Agile Project Framework tailored specifically to complement Scrum - the worlds most widely used team focussed Agile product development approach. It incorporates DSDM's project-focused principles, together with DSDM's rich set of roles and responsibilities that are ideally suited to a corporate project environment. It also offers the robust and fully Agile DSDM techniques for establishing and demonstrating control in a project. At the delivery level, the Agile Project Framework exploits the new simplicity of the DSDM process to work alongside and complement the Scrum product delivery process without the need for a change of approach. AgilePF for Scrum brings together the strength of DSDM at project level and the streamlined simplicity of Scrum at the delivery team level.

One or more aspects of AgilePF may be used to supplement Scrum on a project-by-project basis or an organisation may choose to implement some or all of it as a standard. Regardless, aspects of the AgilePF should be used where they make the use of Scrum easier, more effective or simply make it more accessible to project and organisational stakeholders.

Introducing DSDM

DSDM was initially created in 1994 through collaboration of a large number of project practitioners across many blue chip companies who were seeking to build quality into Rapid Application Development (RAD).

DSDM is a proven framework for agile project management and delivery, helping to deliver results quickly and effectively. It concentrates on strategic goals and incremental delivery of real business benefits while keeping control of time, cost, risk and quality.

Agility is enabled through the encouragement of self-organising, empowered teams working together in a supportive and collaborative manner. Since 1994, DSDM has been effectively applied to a wide range of projects from small software developments all the way up to full-scale business process change.

DSDM agility

DSDM advocates that projects should do just 'enough design up front' 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.

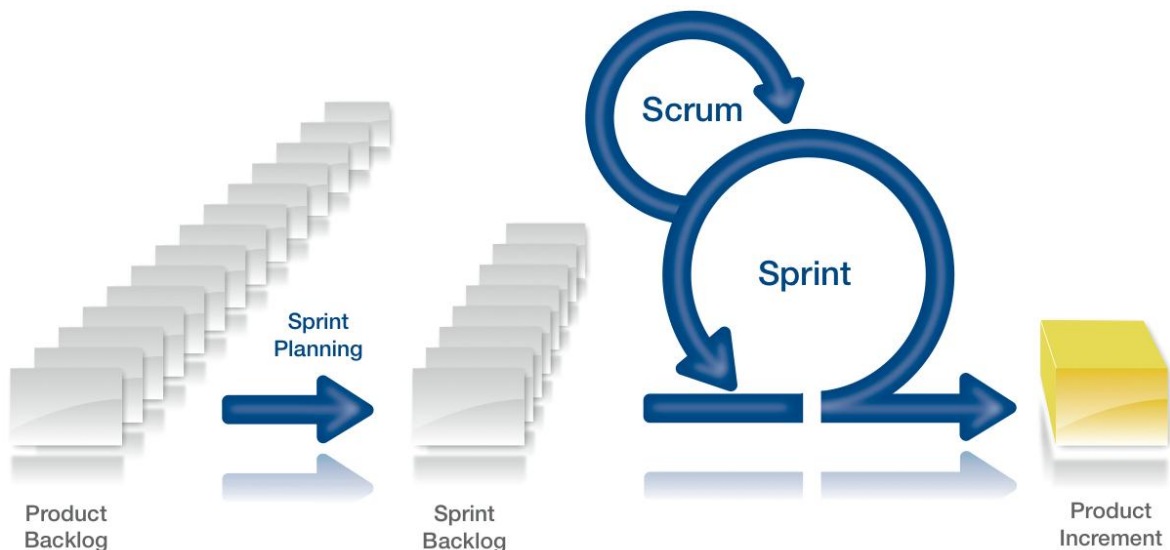
It is important to understand that the Foundations phase of a DSDM project is very different to the Analysis and Design steps in a traditional 'waterfall' approach. In a DSDM project, analysis and design activity carried in Foundations covers the full breadth of the project but deliberately avoids going into detail. Substituting traditional 'big design up front' with DSDM's 'enough design up front' promotes Agility in developing the required solution whilst avoiding the risk of 'no design up front' that makes many larger and more strongly governed organisations so nervous.

(DSDM® and Atern® are Registered Trade Marks of Dynamic Systems Development Method Limited)

Introducing Scrum

Scrum takes an empirical, product-focussed approach to the delivery of products (predominantly software). From a process perspective it is both simple and elegant. Throughout this paper, the authors have tried to use terminology and definition of *what Scrum is* that is consistent with The Scrum Guide 2013 (July 2013, Jeff Sutherland and Ken Schwaber, www.scrumguides.org).

The Scrum Process



- A Product Owner defines an ordered set of requirements for the product to be built. The Product Backlog is a dynamic artefact that evolves over time in line with the changing needs of the business for which the product is being built
- The Scrum Team (comprising the Product Owner, the Development Team and the Scrum Master) use their knowledge, experience and understanding of the product and the requirements to agree a number of the most important items from the Product Backlog to be addressed in a Sprint
- In the process of Sprint Planning, the Scrum Team collaborate to fully understand the backlog items selected for the Sprint and work out how they are going to be implemented in the Product Increment delivered at the end of the Sprint. The two outputs of the Sprint Planning Meeting are:
 - The Sprint Goal – which provides a succinct focus for what the Scrum Team wants to achieve by the end of the Sprint
 - The Sprint Backlog which defines, at varying levels of granularity, the work needed to deliver the Increment
- The duration of the Sprint is agreed at the outset and does not change. Every day, the team holds a 15 minute Scrum meeting. The primary purpose of the Scrum is to allow the Scrum Team to refine their focus on the Sprint Goal, identify any impediments and agree an informal collaborative plan for who will be doing what in the working day up to the next Scrum

- The Sprint ends with a Sprint Review Meeting where the Product Increment is demonstrated and accepted. A Sprint Retrospective follows, in which the team reflect on the effectiveness of the way they worked and agree strategies for improvement where needed.
- A ScrumMaster is present throughout to help keep the Scrum Team focussed on their goal and to help them get the most out of the Scrum Events (Sprint Planning Meeting, Daily Scrum, Sprint Review and Retrospective)

Scrum has a product focus rather than a project focus so more emphasis is placed on incremental release of a product in the context of a product lifecycle than is placed on formally ending development work after an agreed period of time. Effectively, development ends when the value to be delivered by the next Product Increment can not be justified from a business perspective.

AgilePF for Scrum

Philosophy

The DSDM Agile Project Framework embraces the project delivery values at the heart of DSDM and fully aligns with the product development philosophy inherent in Scrum.

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 best achieved when all stakeholders understand and buy into the business vision and objectives, are empowered to an appropriate level, and collaborate in order to deliver a fit-for-purpose solution. This solution will be delivered in the agreed timescale, according to the priorities driven by the business whilst understanding and accepting that change is inevitable as the understanding of the solution grows over time.

Agile Values

Both AgilePF and Scrum fully adopt the values laid out in the Manifesto for Agile Software Development specifically valuing:

Individuals and Interactions over Processes and Tools

In an Agile project, great emphasis is placed on the individual in the context of a collaborative project team. Every individual is expected to be ready, willing, able and empowered 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 should do and how they should do it. The assumption is that the team themselves are best placed to understand what needs to be done and to work out the best way of doing it. AgilePF and Scrum provide appropriate light touch guidance, 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 – that it is for software delivery. However, changing a single word - changing “software” to “solution” - elevates this value from delivery of a software product into the broader context of business change projects.

The main message behind this value is to break the illusion of security and stability that comes from document-driven, predictive 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 now widely accepted to be both wasteful, in terms of time and effort, and ineffective as the basis of governance and control. AgilePF for Scrum embraces the need for high-level versions of requirements, design and planning artefacts in the early phases of the project to frame development and delivery and to support governance. The Agile Project Framework, like Scrum, advocates collaborative techniques with active business engagement to explore detail at the right time and ensure the right solution is delivered. Any documents created as part of the development process should be as light as possible and tied directly to the solution or the work being performed.

Customer Collaboration over Contract Negotiation

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

Looking more broadly at what constitutes a contract – effectively any document signed by those responsible for wanting the solution and those delivering it – it is important to ensure that, where created, all parties follow the principle of such documents being 'light touch' and 'guiding' rather than being 'detailed' and 'prescriptive'. By this definition, the Product Backlog may represent a contract, effectively defining the scope of a project. But it is cast at a high level and requires customer collaboration with less formality to flesh out the detail of requirements throughout the iterative development of the solution during the project lifecycle.

Responding to Change over Following a Plan

This value emphasises the fact that the world around a project is rarely frozen in time. The pace of change in the world of business is now so fast that adopting an approach to building solutions that does not accommodate, or ideally embrace, change is unlikely to lead to a successful outcome. With the level of change that is a fact of life in the modern world, creating detailed, long-term plans becomes a waste of time. In this modern world, the high-level 'light touch' and 'guiding' plans as advocated by AgilePF for Scrum better meet the project, and ultimately the business, need.

The final sentence

AgilePF for Scrum specifically embraces the last sentence in the Manifesto that clearly states, in the context of the values above ***“That is, while there is value to the items on the right, we value the items on the left more.”*** It is important not to shun 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 philosophy.

Principles

The eight DSDM principles underpin AgilePF for Scrum and support the Philosophy. They bring the Agile Values to life by guiding the team in the attitude it must take and the mind-set it must adopt in order to deliver consistently whilst still remaining flexible. Compromising any principle undermines the basic philosophy and introduces risk to the successful outcome of the project.

The eight 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

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, project teams need to:

- Understand the true business priorities
- Establish a valid business case
- Ensure continuous business sponsorship and commitment throughout the project

Specific business roles (described later), in conjunction with the work completed in the Foundations phase, and supported by time-boxed delivery and appropriate prioritisation of requirements, enables teams to fulfil this principle.

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, teams need to:

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

Working in short focussed Sprints, typically of 2-4 weeks, and flexing the scope of what is delivered in that timeframe allows teams to implement this principle and build a reputation for timely and predictable deliveries. Incremental, timeboxed delivery of valuable solution features in short timeframes forms the basis for control over the longer-term delivery of the project.

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, teams need to:

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

The Product Owner role, complemented by the other business roles defined in AgilePF, brings the appropriate subject matter experts into the project so they can contribute to the solution. The ScrumMaster is responsible for facilitating a high level of collaboration between all Scrum Team members. The Business Analyst role also helps ensure collaboration between business and delivery roles is effective.

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 Scrum, it is normal to agree a 'Definition of Done' which define acceptance criteria to be applied generically to all User Stories - the default way of expressing requirements). By default stories are considered 'done' when these generic criteria and the criteria specific to the User Story are met in full.

In order to fulfil this principle, 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, document and test appropriately

The work of the Foundations phase of the project, together with regular reviews throughout the project lifecycle, will help the team to build a solution of the appropriate quality.

In an Agile project everything is tested as early as possible. Test-driven techniques result in a test being written before the deliverable is actually produced. Ensuring that appropriate review and testing activity is carried out in each Sprint ensures that quality of the overall solution is not compromised

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. AgilePF for Scrum advocates establishing a high level understanding of both the scope of the business problem to be addressed and the proposed solution early in the project whilst avoiding wasteful detailed analysis of requirements and detailed solution design

In order to fulfil this principle, teams need to:

- Carry-out appropriate analysis and enough design up front to create strong Foundations
- Formally re-assess priorities and ongoing project viability with each Release

Teams implement this principle using the high level process defined by AgilePF for Scrum, to deliver a solid base of knowledge during Feasibility and Foundations phases. This forms the base for developing incrementally thereafter, with incremental deliveries from one or more Sprints typically contributing to a formal Release.

Principle 6 - Develop iteratively



In order to converge on an accurate business solution, AgilePF for Scrum uses Iterative Development to build the right solution using frequent demonstrations and feedback. The concept of iteration is at the heart of everything developed as part of this Agile 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, 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

Within the constraints of fixed time, cost and quality, change is actively encouraged in order to evolve the most appropriate solution. Iteration and constant review ensure that what is finally delivered is what the business really needs.

Principle 7 - Communicate continuously and clearly



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

In order to fulfil this principle teams will:

- Encourage informal face-to-face communication at all levels
- Hold Scrums (short stand-up team meetings) each working day
- Use workshops, with a facilitator where appropriate
- Use visual communication practices such as modelling and prototyping
- Demonstrate the evolving solution early and often (as a minimum, the end of each Sprint)
- Keep documentation lean and timely
- Manage the expectations of the stakeholders at all levels throughout the project
- Always aim for honesty and transparency in all communication

Agile approaches emphasise the value of human interaction. The roles and responsibilities and communication techniques defined in AgilePF for Scrum encourage team and stakeholder collaboration. Modelling and prototyping allow scrutiny of early impressions of the solution.

These techniques are far more effective ways of communicating than relying on large textual documents, which in themselves provide no financial value to the project or the business.

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

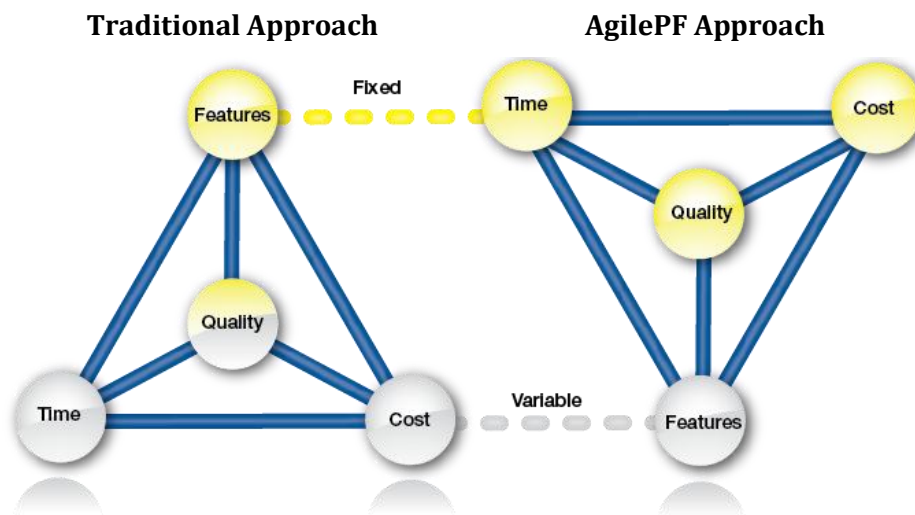
In order to fulfil this principle, the Scrum Team and the Project Manager will:

- 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 planned incremental delivery, which incorporates incremental review and acceptance, based on sound but not detailed planning in the Foundations phase helps all project participants follow this principle.

Variables

Projects have to balance conflicting demands and the four most common demands are: time, cost, features and quality. Trying to fix all four parameters at the outset is impractical and is the cause of many common project problems.



In the traditional approach to project management (shown on the left of the diagram above) the feature 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.

With proper planning, any three of the four project variables can be fixed provided one is allowed to vary. In an Agile project the greatest predictability and the most successful outcome for a project usually comes from fixing time, cost and quality and allowing the scope of the features delivered to vary.

By default, using the AgilePF approach (shown on the right of the diagram above), time, cost and quality are fixed as part of establishing firm Foundations for the project. To reach the point where this can be achieved, an understanding of the high level features is required, sufficient to provide a sensible estimate for those aspects of the project that are fixed. At the same time, it is normal for a subset of the features to be identified as mandatory. This is on the understanding that if the solution does not include these features then it will have either no value or that value will be so severely compromised the project would be considered an outright failure.

When working this way, the contingency within the project comes from the lower priority features, which are progressively re-scoped and, if necessary, de-scoped to protect the commitment to time, cost and quality. The most common options for prioritisation include Absolute Prioritisation where required features are ranked from 1 to n and Relative Prioritisation using a technique such as DSDM's MoSCoW Prioritisation. Solid predictability of end-date for a delivery becomes a reality in projects that use AgilePF.

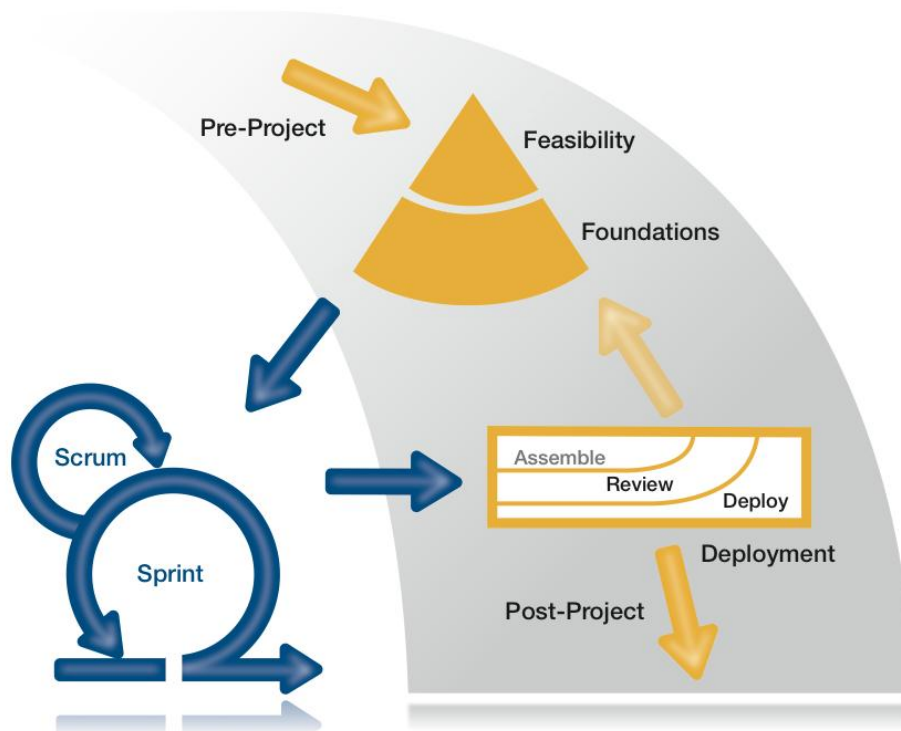
Lifecycle

Every project has a lifecycle that starts with the identification of a potential need and ends at a point when that need is either met or has been rejected. Rejection can occur at any point if the project becomes non-viable from a business perspective.

Traditional project approaches follow a predefined lifecycle, typically executing a sequential pattern of Analyse, Design, Build, Test and Deploy. Larger projects following this approach may execute that pattern a number of times as a means of breaking the project into smaller, more manageable chunks, in order to deliver some of the business value early.

Although Scrum has the concept of a Product Roadmap to provide strategic direction and follows an incremental approach to achieving this, by default it does not capture the full spectrum of the project lifecycle that may be important to many projects and/or organisations where, for example a project encompasses more than just the development of software. Although Scrum can, and often does deal with products beyond software, AgilePF may make these aspects easier to manage in a project context. AgilePF for Scrum takes account of all the work required from initial concept to final measurement of the business benefits realised post-project.

The AgilePF has two elements; one focussed on *project management*, the other on *product delivery*. Scrum deals well with the product delivery and can be readily employed for that aspect of the overall framework. AgilePF deals well with Agile project management. When considering Scrum in the context of AgilePF, it is important to acknowledge the overlap between the project and product delivery elements. The overlap is represented by Scrum's Product Roadmap and Product Backlog which need to reside within the project delivery element of the overall framework which is also responsible for any deployment activity required to get the Product Increment delivered from a Sprint into live use.



The phases in the Agile Project Framework are: Pre-project, Feasibility, Foundations, Evolutionary Development, Deployment and Post-project

The following sections briefly describe these phases.

Pre-Project

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

Feasibility

The Feasibility phase is intended primarily to quickly 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.

Foundations

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 Foundations to review and update the base for the project typically takes significantly less time than the first time Foundations takes place.

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

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

Evolutionary Development

Evolutionary Development takes an iterative and incremental approach to developing the solution as a whole. In the context of a project, the solution will include both the *Product* (often software) and any associated *changes within the business* wanting to exploit that product. Detailed requirements, normally expressed as User Stories are investigated, further analysed and translated into a viable product engineered to full operational readiness.

Iterative Development techniques, either simultaneously or sequentially, address both functional and non-functional requirements including areas such as performance, capacity, security, supportability and maintainability.

Throughout, continued involvement of the Product Owner provides an on-going opportunity to guide development and to validate fitness for purpose of the product. The business roles from AgilePF (the Business Sponsor, Business Visionary and Business Advisor(s)), will often be valuable in a complex corporate environment where it is not feasible for a single individual (Scrum's Product Owner) to understand all the detail of all aspects of requirements for a software product as well as understanding the full implications of the way the business may need to change in order to exploit it. The perspective and input from the other roles may well be valuable to the Product Owner and the rest of the Scrum Team who can rely on the Business Visionary to provide high level context and handle the business change aspects of the wider project and on Business Advisors to provide real depth and detail around specialist requirements such as compliance with specific legislation or working practice.

Deployment

The focus of the Deployment phase is on getting the solution into operational use or ready for market. The number of passes through the Deployment phase will depend the number of Releases scheduled as part of the project. This is, in turn, driven by the business imperatives. Three aspects of deployment considered in this phase are final Assembly (configuration/packaging) of the product, a final Review of what has been delivered and the actual Deployment of the product into operational use.

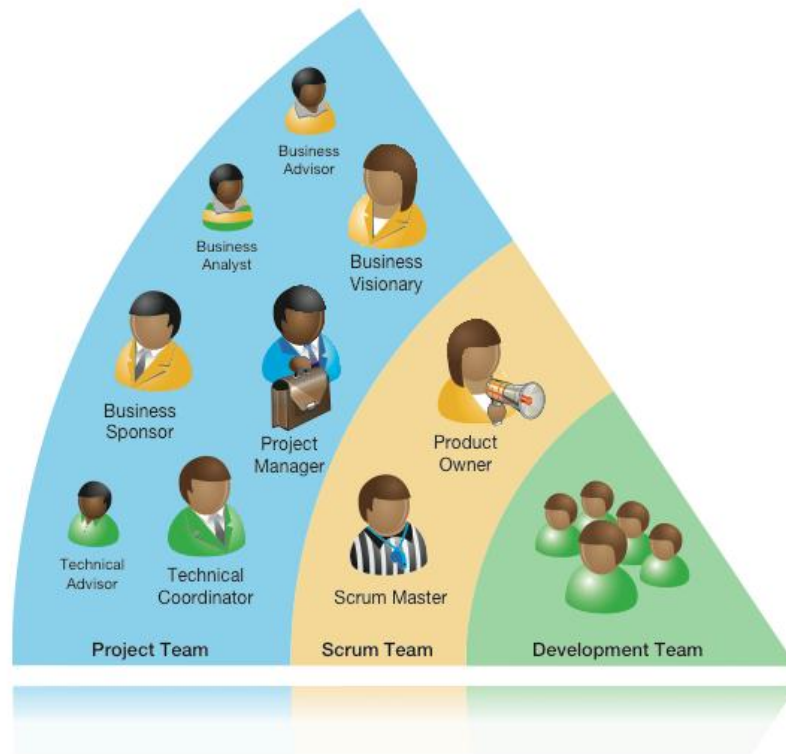
After the final deployment, the project is formally closed. At this point, the whole team hold a retrospective to review the overall project performance, both from the technical and/or process perspective and from the business perspective

Post-Project

After the final deployment for a project, the post-project phase checks how well the expected business benefits have been met.

Roles

Roles in AgilePF for Scrum fit into three categories: Project Team, Scrum Team and Development Team, with the Development Team being a subset of the Scrum Team and the Scrum Team being a subset of the Project Team



The Project Team roles are primarily made up of the roles responsible for strategy and the ‘whole business’ perspective for the project. The Business Sponsor, Business Visionary, Technical Coordinator and Project Manager are collectively responsible for overall project direction, coordination and governance. For larger projects, the Project Team roles are expected to oversee a number of individual Scrum Teams, ensuring that each team remains focussed on the overall business success criteria for the project.

The Business and Technical Advisor roles are typically filled by subject matter experts who advise on matters of detail in a variety of business and technical aspects of solution development. The analytical and facilitative skills of the Business Analyst role provide support to at all levels in the project

The Scrum Team roles (including the Development Team) are responsible for the detailed work involved in defining, developing, testing and perhaps even deploying the solution. The core of the team are engaged throughout the project with other roles, such as Business and Technical Advisors from the Project Team being involved from time-to-time as required to support the Product Owner.

Roles do not necessarily equate to individuals, nor relate directly to job titles. One person may cover multiple roles, or a single role can be shared between several people. Roles organised as described above allow the project to fit more naturally into the broader organisational context.

AgilePF for Scrum roles and their associated responsibilities are described below.

Project Team Roles

Business Sponsor



This role is the most senior Project Team 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.

Responsibilities:

- Owning the Business Case for the project
- Ensuring on-going 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

Business Visionary



This is a senior business role in the Project Team 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 solution once delivered and will be responsible for the realisation of any benefits associated with it. In a pure product context this role will probably be the Product Manager

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 Product Backlog; 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

Project Manager



The Project Manager role is focused on managing the working environment in which the solution is evolving and coordinates all aspects of management of the project at a high level. In line with the AgilePF concept of empowerment, the Project Manager is expected to leave the detailed planning of the actual delivery of the Product Increment to the members of the Scrum Team. Managing an empowered team requires a facilitative style rather than a command and control style. 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 establishing the foundations of the project through to the deployment of the solution.

Responsibilities:

- Ensuring effective and timely communication with the senior stakeholders and project governance authorities with the agreed and appropriate level of frequency and formality
- Performing high level project planning and scheduling, but **not** Sprint planning.
- Monitoring progress against the baselined project and increment plans
- Managing risk and any issues as they arise, or are escalated from the Scrum Team(s), collaborating with senior business and/or technical roles as required to ensure resolution
- Motivating and ensuring empowerment of the Scrum Team(s) to meet their objectives
- Attending Scrum meetings, as an observer, to keep a current understanding of progress and issues
- To ensure that any important external issues that the Scrum Team(s) need to be aware of are brought to their attention

Technical Coordinator



The Technical Coordinator performs the same function, from a technical perspective, as the Business Visionary does from a business perspective. He/she ensures that Scrum Teams work in a consistent way within a project and deliver compatible output and also ensures that the project meets the desired technical quality standards appropriate to both the project and, where applicable, the wider product or organisation.

This role provides the glue that holds the project together, providing advice and guidance on technical decisions and innovation. A framework of empowerment ensures that this is a guiding role rather than a commanding one, with the Scrum Teams empowered within defined boundaries to collaborate innovate and manage for themselves how they go about building the solution

Responsibilities:

- Predominantly during the Foundations phase:
 - Advising on the achievability of non-functional requirements
 - Working with the business roles and the Business Analyst to evaluate the technical options and, at a high level, decide the best way to fulfil the business needs
 - Determining the technical environments
- Predominantly during the Evolutionary Development phase:
 - Agreeing and controlling the technical architecture
 - Advising on and coordinating each team's technical activities
 - Identifying and owning architectural and other technically based risk
 - Promoting appropriate standards of technical best practice
 - Controlling the technical configuration of the solution
 - Empowering the technical roles within the Development Team(s), to appropriate levels, within their responsibilities
 - Acting as the final arbiter of technical differences between Development Team members
- Predominantly during the Deployment phase:
 - Approving the solution as technically fit for purpose prior to deployment
 - Managing technical aspects of the transition of the solution into live use

Business Advisor



The Business Advisor supports the Project Team and Product Owner by providing specialist, and where appropriate detailed, input to requirements, development and testing. They may be an intended user or beneficiary of the solution or they may provide guidance on areas of compliance.

The Business Advisor supports the Product Owner through the provision of detailed or specialist input and advice with regards to:

- Requirements, design and review sessions
- Business perspective for day-to-day decisions
- Business scenarios to help define and test the solution
- Assurance that the solution is evolving correctly
- Business acceptance testing
- Development of business user documentation
- User training

Technical Advisor



The Technical Advisor supports the Project Team and Product Owner by providing specific and/or specialist technical input to the project often from the perspective of those responsible for operational change management, operational support, on-going maintenance of the solution etc.

From their specialist technical perspective the Technical advisor may help support:

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

Business Analyst



The Business Analyst supports the Project Team throughout the Feasibility and Foundations phases, typically doing most of the investigation and high-level analysis work that is required along with the preparation of many of the documentary products that may be delivered from those phases.

During the Evolutionary Development phase their analysis and facilitation skills may be used to support the Product Owner in ensuring requirements are properly expressed, form a coherent whole and are properly understood by the Development Team.

In some cases, where the Product Owner has all the required skills the Business Analyst role may be subsumed into the Product Owner role and the Product Owner may take on the Business Analyst responsibilities in supporting the Project Team roles.

Responsibilities where appropriate and valuable:

- Assisting the Business Visionary in the formulation and promotion of the Business Vision
- 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 Product Owner to formulate and communicate solution options
- Working with the Project-Team 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 are of good quality and are analysed and managed appropriately
- 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
- Liaising with the Business Visionary in organising support for the solution through implementation into live use

Scrum Team Roles

Scrum Master (replaces AgilePF Team Leader)



The Scrum Master is responsible for ensuring Scrum is understood and is properly applied. Often described as a servant-leader, the Scrum Master encourages and supports, the self-organising team in achieving their potential, and the wider organisation in enabling that to happen.

From the Development Team's perspective, the Scrum Master is there to:

- Help them self-organise
- Help them apply Scrum properly and effectively
- Help remove impediments to both progress and the application of Scrum

From the Product Owner's perspective, the Scrum Master is there to help with techniques for:

- Shaping and ordering the Product Backlog
- Communicating the vision, goals and intended use of the product to the Development Team
- Describing Product Backlog items to the Development Team clearly and unambiguously

For the benefit of all, the Scrum Master is there to help the Project Team, and the wider organisation, understand the philosophy and practicalities of Scrum and the AgilePF extension to Scrum, so that the combination can be used effectively for the intended product development work.

To help achieve the objective of optimising the use of Scrum, the Scrum Master is likely to act as a facilitator for the Scrum events of Sprint Planning, the daily Scrum, the Sprint Review and the Sprint Retrospective. The Scrum Master roles also takes on the AgilePF DSDM Coach responsibilities.

Product Owner (replaces AgilePF Business Ambassador)



The Product Owner is responsible for ensuring that the Evolving Solution created by the Development Team is properly aligned with the Business Vision, that the final product delivered by the Scrum Team represents the best value for the sponsoring business and for ensuring that the Product Increment delivered at the end of each Sprint represents the most valuable step in that direction.

In an AgilePF context, the Product Owner generally comes from the business area that will be the beneficiary of the project and/or represents an end-user of the solution who is outside the organisation running the project. The Product Owner provides detailed business information on any matter from the perspective of those who will ultimately use the solution.

The role is responsible for the informal day-to-day communication channels between the team and the wider business users and beneficiaries of the solution.

In order to provide unambiguous direction for the Development Team, it is important that the Product Owner is a single individual not a group. In practical terms, other stakeholders and domain experts may interact with the Scrum Team, as complex messages passed via an intermediary are likely to be misinterpreted, but the Product Owner always 'owns' the requirement being addressed. Only one Product Owner can define work for the Development Team and the Development Team can only accept work from one Product Owner.

Responsibilities:

- Ensures that the product delivered by the team has optimal business value:
 - Shapes the Product Backlog, in line with the Business Vision, to ensure the most valuable aspects of the solution are developed and delivered as early as is practical
 - Contributes to requirements, design and review sessions
 - Provides a business perspective for day-to-day decisions
 - Describes business scenarios to help define and test the solution
 - Provides day to day assurance that the solution is evolving correctly
- Helps optimise the transition of the solution to live business use:
 - Communicates with business users outside the project.
 - Organises business acceptance testing as required
 - Develops business user documentation as required
 - Ensures adequate user training is carried out

Development Team

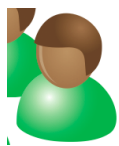


Scrum is explicit in its guidance that there should be no demarcation of roles and responsibilities within the Development Team. The Scrum Guide states “Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule” and “Development Teams do not contain sub-teams dedicated to particular domains like testing or business analysis”

AgilePF guidance contradicts Scrum primarily for pragmatic reasons. In an ideal world every member of the Development team will be a very experienced, highly competent person capable of working across all technologies involved in the project and with the desire and capability to perform all the analysis, design, development and testing work required to build a solution. In many larger organisations this ideal is not a current reality and the practicalities of making it a reality would take years – and that assumes there is a real desire to make the change and the budget to make it a reality.

In an organisation where demarcation of roles already exists (and only in that circumstance) the following definitions may be helpful in selecting the right blend of individuals for a given Development Team. Even in these circumstances other key aspects of guidance on the Development Team must be a reality e.g. “They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality” and “Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole”

Solution Developer



This role provides a development perspective within the context of a collaborative Development Team, recognising that in many corporate environments, a developer is a skilled specialist distinct from a tester. The Solution Developer interprets business requirements and translates them into a deployable solution that meets functional and non-functional needs.

Responsibilities:

- Works collaboratively with Product Owner and other Solution Developers and Testers
- Undertakes Iterative Development of the deployable solution
- Adheres to technical constraints laid out in the Solution Foundations
- Participates in quality assurance work to ensure products are fit for purpose
- Tests their own output prior to independent testing

Solution Tester



This role provides a testing perspective within the context of a collaborative Development Team, recognising that in many corporate environments, a tester is a skilled specialist distinct from a developer. The Solution Tester designs, builds and performs all types of testing during development and deployment of the solution, focusing predominantly on the technical aspects of testing.

Responsibilities:

- Works collaboratively with Product Owner, Solution Developers and other Solution Testers
- Carries out technical testing
- Reports test results to the Technical Co-ordinator informed of depth / coverage of testing (for Quality Assurance purposes)
- Assists Product Owner and Business Advisor roles to carry out important business related testing

Other Roles from AgilePF

Workshop Facilitator

The Workshop Facilitator manages the workshop process, ensuring the necessary preparation and communication happens. The role is responsible for the workshop context, but not the content and as such has no stake in the workshop outcome, just that the desired outcome is reached and accepted.

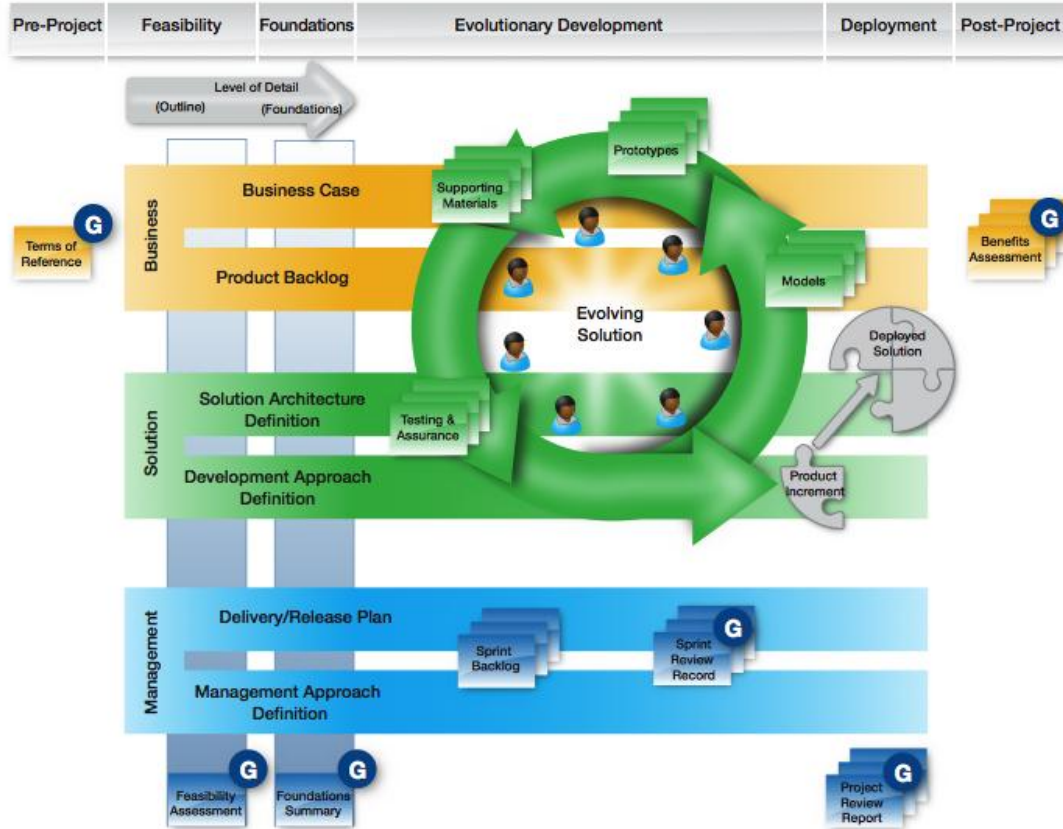
DSDM Coach

In the AgilePF for Scrum, this role is part of the Scrum Master role

Products (Artefacts)

The Agile Project Framework defines a total of 14 products including the solution being created. Many of these are documents that should be created where they add value. The use and formality of these artefacts may vary from project-to-project but all should be actively considered for every project.

Some products have a role to play in project governance – these are labelled **G** in the diagram below.



Terms of Reference

Defines at a very high level the objectives and business drivers for the proposed project.

Business products

The business products are created in outline during Feasibility, and refined to provide firm and enduring foundations for the project as a whole by the end of the Foundations phase without becoming too detailed and restrictive. The Business Foundations normally comprises a Business Case and the Product Backlog. Both may continue to evolve throughout the project

Business Case

Describes essential business considerations that justify the project, and then are used to assess the viability of the project moving forwards.

Product Backlog

The Product Backlog replaces the standard AgilePF Prioritised Requirements List. It provides a set of high-level User Stories that the project needs to address, indicating their priority with regards to meeting the overall project objective.

Solution products

The solution products are created in outline during Feasibility and refined to provide firm and enduring foundations for the project as a whole by the end of the Foundations phase without becoming too detailed and restrictive. The Solution Foundations normally comprises a Solution Architecture Definition and a Development Approach Definition. Both may continue to evolve throughout the project

Solution Architecture Definition

Provides an overview and architectural framework for both business and technical aspects of the potential solution. This will evolve as the project proceeds.

Development Approach Definition

Defines the standards and practices to be adhered to and provides guidance on how the solution should be evolved as the project proceeds. It includes the *'Definition of Done'*.

Management products

The management products are created in outline during Feasibility and refined to provide firm and enduring foundations for the project as a whole by the end of the Foundations phase without becoming too detailed and restrictive. The Management Foundations normally comprises a high-level Delivery/Release Plan and a Management Approach Definition. Both may continue to evolve throughout the project

Delivery/Release Plan

Provides an initial high-level schedule of Sprints, Releases and other activities for development, testing and deployment of the solution. For larger projects a single high-level Delivery Plan will deal with coordination of the efforts of multiple Scrum teams. On smaller simpler projects the Delivery/Release Plan may be integrated with the Product Backlog with User Stories identified as belonging to a particular planned release. This plan is constantly reviewed and revised as the project progresses to reflect the latest business demands and predicted outcomes in terms of timescales and delivered scope

Management Approach Definition

Describes the approach to the set-up and management of various aspects of the project, including how the project will be organised and governed. It also describes the approach to managing Change, Configuration, Communication and Risk.

Feasibility Assessment

The Feasibility Assessment provides a snapshot of the evolving business, solution and management products described above as they exist at the end of the Feasibility phase. Where created, 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 described or as an executive summary covering the key aspects of each of them.

Foundations Summary

The Foundations Summary provides a snapshot of the evolving business, solution and management products described above as they exist at the end of the Foundations phase. Where created, 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 Foundations Summary may be expressed as a baselined collection of the products described or as an executive summary covering the key aspects of each of them.

Sprint Backlog

The Sprint Backlog defines the work the Development Team will perform to turn Product Backlog items into a "Done" Product Increment. It is a forecast by the Development Team about what functionality will be in the next Product Increment and the work needed to deliver that functionality. The Sprint Backlog is kept under constant review and revised as the Sprint progresses. In many cases the Sprint Backlog will take the form of a Team Board with User Story cards and Task cards on it along with a clear indication of who in the Development Team is doing what and any blockers or other issues currently requiring attention.

Sprint Review Record

Where useful, the Sprint Review Record captures what has been achieved in the Sprint and any feedback, which will influence plans moving forward. Any outstanding issues are considered in the context of the Delivery/Release Plan and future Sprint plans. If review points are built into the Sprint, for example as part of the AgilePF Timeboxing technique (described later), the review record should

be used to capture feedback at each review point and will therefore evolve as the Sprint progresses. This record may be particularly useful in more regulated environments where it is important to demonstrate compliance with specific legislation or process controls as it will provide an audit trail of reviews that can be focussed on subjects of interest.

Evolving Solution

The Evolving Solution 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.

At the end of each Sprint the Product Increment is a demonstrable baseline of the Evolving Solution. Such a baseline deployed into live use and becomes the Deployed Solution.

Project Review

An evolving artefact updated after a whole project retrospective at the end of every Release to reflect project performance and learning.

Benefits Assessment

Describes how the benefits predicted in the Business Case have actually accrued as the Deployed Solution has been used.

DSDM Agile Project Framework Techniques

Introduction

The following AgilePF practices are proven techniques that may be used by the project team to help fit practically and predictably into a governed, corporate project, programme or portfolio management environment without compromising Agility. This allows AgilePF to provide the appropriate level of scalability and rigour, whether the project is small and simple or large and complex.

Facilitated Workshops and Modelling represent techniques that may be usefully employed in any project, Agile or otherwise. Both these techniques promote and support communication and collaborative working.

MoSCoW Prioritisation offers a project focussed 'wrapper' for the simple ordering of a Product Backlog. Timeboxing offers a structural overlay to a Sprint that may help improve predictability of delivery and also provides a structure that may help where Product Owner availability is limited

Facilitated Workshops

Facilitated Workshops encourage collaborative working and enable high quality team-based decisions to be made in a shortened timeframe. People brought together as a group communicate more effectively and generate more creative solutions.

A well-run Facilitated Workshop also delivers an outcome with a high degree of buy-in and ownership from those people who have taken part.

Ideally, workshops are independently facilitated by someone external to the project. At the very least, the Workshop Facilitator should be independent of the workshop result, to ensure that all ideas and contributions are given equal weight. A trained facilitator creates an environment that allows full participation.

Effective workshops follow a well-defined and carefully thought-out process. This process should include defining the objective, identifying appropriate participants, creating an agenda, managing the logistics and distributing any pre-reading to participants.

Facilitated Workshops are particularly valuable when applied to activities such as requirements identification and refinement, prioritisation, Release or Sprint planning, risk analysis, problem solving, product reviews and retrospectives.

Modelling

Modelling is a technique for collaboratively evolving diagrams and pictures that define the problem or intended solution. They are used in Agile projects to improve communication through visualisation.

How often models are used and their formality depends on the nature of the project and the team's level of skill and experience in modelling techniques.

Models will also vary depending on the type of project, prevailing standards and best practice. In deciding what models should be created and when, the simplest rules to follow are:

- Be able to justify the value of the model for enhancing understanding of the given subject
- Use an approach that works for you and your organisation
- Do enough and no more so that the purpose of the model is achieved.

MoSCoW Prioritisation (optionally applied to the Product Backlog)

Delivering on a guaranteed date to a fixed cost means that some work originally planned for a delivery may have to be deferred. It may also be necessary to include work not originally identified.

Essential work must be completed and it is only less critical work that may be omitted from a delivery.

MoSCoW Prioritisation is a straightforward technique that can be used to achieve this, by prioritising the Product Backlog using these MoSCoW rules:

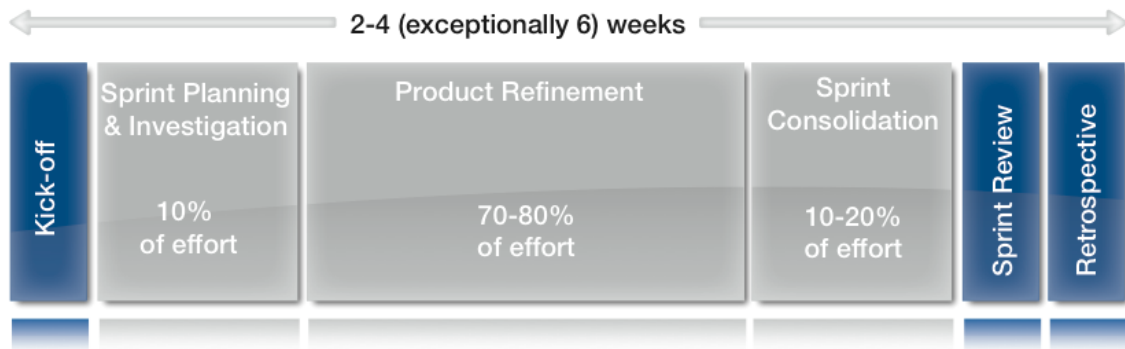
- **Must Have:** requirements that are fundamental to the solution. Without these the solution will be unworkable and useless. Must Haves define the Minimum Usable SubseT which the project *guarantees* to deliver (provided the estimated effort associated with these requirements is less than 60%)
- **Should Have:** important requirements for which there is a short-term workaround. Normally classed as mandatory when more time is available, but without them the business objective will still be met
- **Could Have:** for lower value requirements that can more easily be left out
- **Won't Have this time:** for requirements that can be included in later development. Won't Haves are excluded from plans for the current delivery

MoSCoW Prioritisation can provide the basis for decision making about project team activity at all levels. The advantage of MoSCoW is that it allows business expectations to be set at the Project level, in the knowledge that the team will definitely deliver the Must Haves, they are likely to deliver most or all of the Should Haves, and they may deliver some Could Haves. For more information on MoSCoW prioritisation refer to the DSDM Agile Project Framework Handbook – available as a book or free to view electronically at www.dsdm.org.

DSDM Timeboxing (optionally applied to a Sprint)

The following is a representation of the AgilePF Timebox applied to a Scrum Sprint. Applying some or all aspects of the Timeboxing technique may help the team better control their Sprint and help them translate a delivery aspiration for a Sprint into a commitment to deliver what they promised when they promised it. I.e. a potentially releasable increment of the product to the agreed level of quality and meeting the most valuable of the requirements identified for it.

All of the activity described in the steps below is required for any Sprint. The steps provide some structure that may help the team control the work they do. It is important for the team to consider the challenges they face and to decide which, if any, of these distinct steps need to be put in place and how much effort should be associated with each.



Kick-Off

In the Kick-off session, the Team agrees what will be taken from the Product Backlog for this Sprint.

Sprint Planning and Investigation

This step provides a firm foundation for work to be carried out during the Product Refinement that follows. At this time, the Scrum Team achieves an understanding of the detail of the requirements to be met and how they will be addressed as part of the Evolving Solution. The Sprint Backlog of detailed work to be completed is baselined at the end of this part of the Sprint and the team commits to deliver one or more products by the end of the Sprint, based on their detailed investigation and their estimates of the work required to do this.

Where requirements are already well understood – perhaps because they have been explored as part of the work of the previous Sprint – the effort will be in line with the standard equating to 2 hours per week of Sprint. Where little or no understanding of the requirement exists, the team may spend double that (10%) or perhaps even more of the duration of the Sprint in this step.

At the end of Investigation, all Scrum Team members and, where it will be useful, other stakeholders should attend a review. The purpose of the review is to ensure that the Team and any stakeholders attending the review have a shared understanding of what the team plan to do to address the requirements for the Sprint

Product Refinement

This is where most work is carried out in the Sprint and development work and associated testing should be virtually finished by the end of this period.

At the end of Product Refinement there is a major review to look at which deliverables have been created and see what amendments will be needed to satisfy the acceptance criteria. The Product Owner and anybody else involved in product acceptance should attend in order to provide their feedback and help prioritise the work to be completed in Sprint Consolidation.

When Product Refinement is complete, no new work should be started and any remaining work to complete in-progress deliverables should be carefully prioritised for Sprint Consolidation.

Sprint Consolidation

Actions agreed at the end of the Refinement review are carried out together with any final work required to satisfy organisational or project standards. Final testing is completed and any product failing to pass its tests is not considered to be delivered. The effort associated with Sprint Consolidation will depend on how well honed technical and quality practices are within the team and in the complexity of the overall solution configuration. E.g. when delivering a straight forward ecommerce application using modern technologies such as .Net, with good technical disciplines within the team Consolidation may just be an hour or two. Where the team is challenged with legacy integration issues or where technology, skill or environment doesn't readily support Test Driven Development and Continuous Integration up to 20% of the Sprint duration may be needed if a potentially releasable Product Increment is to be delivered.

Sprint Review

The Sprint review often takes the form of a demonstration of the Product Increment delivered by the Scrum Team. It looks formally at what the Team delivered in the Sprint and either acknowledges that the increment of the product delivered is fit for purpose or highlights where it is not. Any requirements that were intended to be addressed but were for some reason de-scoped are also noted.

Anything planned for delivery but not actually delivered may be considered for inclusion in a later Sprint at the next planning session.

Sprint Retrospective

The Sprint Retrospective provides the opportunity for the Team to explore the effectiveness of the way they worked in the Sprint. Particular attention is paid to:

- Any issues with quality or anything else that lead to the solution or aspects of it deemed not fit for purpose.
- Understanding the reasons behind why any requirements were de-scoped

Where appropriate, a summary of what was delivered and formal acceptance of it, along with any notable shortfalls or deficiencies may be captured in a Sprint Review Record.

In Summary

The DSDM Agile Project Framework for Scrum offers teams and organisations a range of options to help with some of the challenges of scaling and governance for Scrum projects. The over-arching process framework, roles, products and techniques have all evolved over time as part of DSDM and have been repeatedly proven in corporate environments. Far from being heavy and prescriptive, the framework as it applies to Scrum is intended to provide a 'toolkit' for teams aimed at making their lives simpler and easier. This is achieved by offering a fully Agile structure to support larger projects and projects in organisations accustomed to a more formal approach to project management and governance.

About the Authors

All the authors are actively engaged in consultancy, training and coaching of Agile methods including DSDM and Scrum. Collectively they have decades of experience with the application of Agile in both public and private sectors and have worked with organisations ranging from software houses employing just a handful of people to some of the biggest and most successful multi-national companies in the world.

Andrew Craddock andrew.craddock@nlightentraining.com	Barbara Roberts barbara@roberts66.co.uk
Julia Godwin jdg@be-consulting.co.uk	Keith Richards k.richards@agilekrc.com
Dorothy Tudor dottudor@tcc-net.com	

Thanks also for feedback and review from members of the Scrum Community:

Jose Casal-Gimenez (jose.casal@eveningcoders.com), Nick Clare (nick.clare@ntlworld.com),
Mischa Ramseyer (mischa@pragmatic-solutions.ch)