

Agile: An Executive Guide

Real results from IT budgets

Second edition

Jamie Lynn Cooke

IT Governance

Agile: An Executive Guide

Real results from IT budgets

Second edition

Copyright IT Governance Ltd

Provided by IHS Markit under license with IT Governance Ltd. See=/, User=,

No reproduction or networking permitted without license from IHS.



Documents

Document Sharing Hub

Full document details: ITG 2016

Agile: An Executive Guide

Real results from IT budgets

Second edition

JAMIE LYNN COOKE



IT Governance Publishing



Documents

Document Sharing Hub

Copyright IT Governance Ltd

Provided by IHS Markit under license with IT Governance Ltd. See /, User=, No reproduction or networking permitted without license from IHS.

No reproduction or networking permitted without license from IHS.

Full document details: ITG 2016

Every possible effort has been made to ensure that the information contained in this book is accurate at the time of going to press, and the publisher and the author cannot accept responsibility for any errors or omissions, however caused. No responsibility for loss or damage occasioned to any person acting, or refraining from action, as a result of the material in this publication can be accepted by the publisher or the author.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form, or by any means, with the prior permission in writing of the publisher or, in the case of reprographic reproduction, in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publisher at the following address:

IT Governance Publishing
IT Governance Limited
Unit 3, Clive Court
Bartholomew's Walk
Cambridgeshire Business Park
Ely
Cambridgeshire
CB7 4EA
United Kingdom
www.itgovernance.co.uk

© Jamie Lynn Cooke 2011, 2016

The author has asserted the rights of the author under the Copyright, Designs and Patents Act, 1988, to be identified as the author of this work.

First published in the United Kingdom in 2011
by IT Governance Publishing.

ISBN 978-1-84928-190-4



Documents

Document Sharing Hub

Copyright IT Governance Ltd

Provided by IHS Markit under license with IT Governance Ltd. See=/, User=,

No reproduction or networking permitted without license from IHS.

Full document details: ITG 2016

Second edition published in 2016 by IT Governance Publishing.

ISBN: 978-1-84928-796-8



DEDICATION

To my sister, Susan, for being a constant inspiration in my life.

.....

FOREWORD

Over the past two decades, Agile methods have enabled IT projects to deliver unprecedented levels of business value to organizations worldwide. Some of these benefits can be obvious from the first time that these approaches are used in an organization: higher quality outputs, more relevant software solutions, greater levels of user satisfaction, and tighter control of ongoing IT expenditure.

What may be less evident, however, are the compounded benefits that Agile methods can bring to organizations in the longer term: most notably, the establishment of an organizational culture that cultivates quality, inspires greater levels of employee motivation, and encourages staff to continually deliver high business-value outputs.

Agile: An Executive Guide gives you the bottom-line information that you need to harness the business-value benefits of Agile within your organization, and generate significantly greater returns on your IT investment.

Dr Jonathan Gray

Senior Researcher and Project Leader,
e-Government

National ICT Australia Limited (NICTA)

PREFACE

How often are the software development projects in your organization delivered on time? Within budget? Most importantly, how much is it costing you to *maintain* and *support* your delivered software to address usability issues, quality issues, or the inability of the delivered software to genuinely meet the current (and future) needs of the business?

It is no secret that information technology (IT) projects have earned a notorious reputation for budget overruns, missed deadlines, lack of alignment with the business, and dissatisfied users. These outcomes are so commonplace that they have become an expected part of the industry. (Just ask commercial software vendors how much they inflate their up-front development costs to allow for the inevitable overhead of addressing software problems during the warranty period?)

Arguably, the most quantifiable indicator of escalating software issues is the fact that, for many organizations, the ‘support desk’ has ballooned from a few part-time staff members to a fully staffed standalone department. This increase in the need for software support is not just due to the increasing number of IT systems and users; it is a symptom of software solutions which are being released with:

- fewer quality controls (bug fixes)
- less intuitive interfaces (phone and online support)

Preface

- a lack of alignment with the true business needs of the stakeholders (enhancements).

It does not matter whether your organization is a commercial software vendor, a 40-person IT consulting firm, or a large multinational with a dedicated IT department for internal business systems; whether you work in the private, not-for-profit or public sector; whether you are a recent start-up or an established company which has been in the industry for over 50 years. No organization is immune to these software issues. Which is exactly why Agile methods emerged in the first place.

Since the 1990s, Agile methods have provided the IT industry with an approach to software development that focuses on the quality, relevance and extensibility of software solutions. These approaches overcome traditional software development issues by positioning IT resources to regularly deliver *high business-value outcomes*; they replace up-front investment in software solutions with *incremental investment* based on proven business-value returns; most critically, they *protect the organization* from wasting money on software solutions that will only need to be reworked or replaced in the future because they do not meet the needs of the intended recipient.

In picking up this book, you have likely traded off your valuable time with one primary goal in mind – to decide whether Agile methods are worth pursuing in your organization – and, if so, to determine the best way to introduce, incorporate and leverage these approaches in your current environment.

Preface

Accordingly, this book has been written with only one objective: to provide IT executives and managers with the information that they need to make realistic (and justifiable) decisions on whether Agile is right for their organization, and the strategies to make it happen.

ABOUT THE AUTHOR

Jamie Lynn Cooke has 25 years of experience as a senior business analyst, project manager and solutions consultant, working with more than 130 public and private sector organizations throughout Australia, Canada and the United States.

She is the author of *Agile Productivity Unleashed: Proven Approaches for Achieving Real Productivity Gains in Any Organization*, a book written specifically to explain Agile methods in non-technical business terms to managers and executives outside of the IT industry; *Everything you want to know about Agile: How to get Agile results in a less-than-Agile organization*, which gives readers strategies for aligning Agile work within the reporting, budgeting, staffing, and governance constraints of their organization; and *The Power of the Agile Business Analyst: 30 surprising ways a business analyst can add value to your Agile development team*, which details 30 core activities that Agile business analysts can undertake to ensure that Agile development teams deliver the highest possible business value solutions for the organization.

She is also a well-regarded speaker on both business and technology topics, most recently presenting on issues such as: *Getting Management and Customer Support for Using Agile* and *When is Agile **Not** the Answer?* at the Business Process Modelling world conference in Brisbane, Australia and at the AgileCanberra professional forums.

About the Author

Jamie has been working hands-on with Agile methods since 2003, and has researched hundreds of books and articles on Agile topics. She is a signatory to the Agile Manifesto, has attended numerous Agile seminars; and has worked with prominent consultants to promote Agile methods to numerous organizations worldwide.

Jamie has a Bachelor of Science in Engineering Psychology (Human Factors Engineering) from Tufts University in Medford, Massachusetts; and a Graduate Certificate in e-Business/Business Informatics from the University of Canberra in Australia.

ACKNOWLEDGEMENTS

My continued thanks to the pioneers and thought leaders of the Agile world, most notably Kent Beck, Martin Fowler, Alistair Cockburn, Jeff Sutherland, Mike Cohn, Ken Schwaber and Jim Highsmith, for their passionate work in developing and refining Agile methods over the past two decades.

Thanks also to the small and large organizations worldwide that have allowed their experiences in using Agile to be shared with others, including Nokia Siemens Networks™, Yahoo!™, Google™, Microsoft™ and BT™.

Special thanks to Vicki Utting and Angela Wilde of IT Governance Publishing, Neil Salkind of the Salkind Literary Agency, Stacey Czarnowski of Studio B and Liz Gill at LCI Proofreading & Editing Services for their ongoing support and sage advice.

Many thanks, as well, to the people who taught me the most about the strategies of the business world over the past 25 years, especially Roland Scornavacca, Tony Robey and Peter Walsh; to Rowan Bunning for being an unending source of Agile knowledge; to Dr. Jonathan Gray of NICTA for his industry insights; and to the writers and teachers who inspired me, particularly Richard Leonard¹ for his amazing ability to encourage writers with his humor and enthusiasm.

...¹ Richard Leonard's website: www.richardleonard.net

Acknowledgements

Finally, my eternal gratitude to my parents, my US family, my Australian family, and my friends, most especially Michele, Susan, Elissa, Janice and Linda, for continuing to be my sanity check in this world. Most of all, thank you to my husband, David, for 24 years of love and laughter.

CONTENTS

Chapter 1: An Executive Brief on Agile.....	16
What is Agile?	16
A two-minute history of Agile	19
The 10 core business benefits of Agile	24
Common Agile methods	26
Who uses Agile?	55
Chapter 2: What Can Agile Do For My Organization?	58
Agile is not for every organization	58
Agile is not for every project	60
Dispelling Agile myths	61
Your Agile ROI	65
Your organizational culture	69
Your decision.....	73
Chapter 3: Five Steps to Agile Success.....	75
Step One: Choosing the right kick-off point ...	75
Step Two: Avoiding common traps	85
Step Three: Establishing your baseline	88
Step Four: Monitoring your investment	93
Step Five: Expanding Agile	98
More Information on Agile	103
ITG Resources.....	110



CHAPTER 1: AN EXECUTIVE BRIEF ON AGILE²

What is Agile?

'Agile' is a collective term for methods (and practices) that have emerged over the past two decades to increase the relevance, flexibility and business value of software solutions. These approaches are specifically intended to address the problems that have historically plagued software development

activities in the IT industry, including budget overruns, missed deadlines, low quality outputs, and dissatisfied users.

Agile methods are common-sense approaches for applying the finite resources of an organization to deliver high business-value software solutions.

Although there is a broad range of Agile methods in the IT industry – from software development and project delivery approaches to strategies for software maintenance – all of these methods share the same basic objectives:

- replace large up-front investment in software solutions with *incremental investment* based on business value returns

² For those who follow this author's writing, some of the introductory material from *Agile Productivity Unleashed* and *Everything You Want to Know About Agile* has been adapted for use in this book, serving the same purpose as the original.

1: An Executive Brief on Agile

- focus project team members on delivering capabilities that generate the *highest business value* for the organization
- encourage *ongoing communication* between the business areas and project team members to increase the relevance, usability and quality of delivered software
- entrust and empower staff to *deliver value*
- position software solutions to be *responsive* to ongoing industry, organizational and technology changes.

Some of the most common Agile methods include:

- iterative strategies for managing software development projects, such as *Scrum*, *Dynamic Systems Development Method (DSDM)*, *Feature-Driven Development™ (FDD™)*, the *Rational Unified Process® (RUP®)*, and the *Agile Unified Process (AUP)*
- strategies for optimizing software development work, such as *eXtreme Programming (XP™)*, and *Lean Development*
- strategies for managing software development projects, as well as maintenance and support activities, such as *Kanban* and *Scrumban*
- extensions of Agile methods to support large enterprise-wide teams and shared corporate objectives, such as the *Scaled Agile Framework® (SAFe®)*, *Scrum of Scrums*, *Large-Scale Scrum Framework (LeSS)* and *Nexus*.³

³ Further detail on these methods is provided in the [Common Agile methods](#) section of this chapter.

1: An Executive Brief on Agile

These Agile methods have been (and continue to be) successfully used by thousands of organizations worldwide⁴, most notably in the United States, Europe and Australia. Some of the more prominent organizations using Agile methods include Nokia Siemens Networks^{TM5}, Yahoo!^{TM6}, Google^{TM7}, Microsoft^{TM8}, BT^{TM9}, Bankwest^{TM10}, SunCorp^{TM11}, and Wells Fargo^{TM12}.

⁴ As evidenced by the number of signatories to the Agile Manifesto (www.agilemanifesto.org) as at December 2015.

⁵ NokiaSiemens and Agile Development, Haapio P, JAOO (2008): http://jaoo.dk/file?path=/jaoo-aarhus-2008/slides//PetriHaapio_CanAGlobalCompany.pdf.

⁶ Lessons from a Yahoo! Scrum Rollout, Mackie K (2008): <http://campustechnology.com/articles/2008/02/lessons-from-a-yahoo-scrum-rollout.aspx>.

⁷ Scrum Tuning: Lessons Learned at Google, Sutherland J (2006): https://www.youtube.com/watch?v=9y10Jvruc_Q.

⁸ Microsoft Lauds Scrum Method for Software Projects, Taft D K (2005): www.eweek.com/c/a/IT-Management/Microsoft-Lauds-Scrum-Method-for-Software-Projects/.

⁹ Agile Coaching in British Telecom, Meadows L and Hanly S (2006): www.agilejournal.com/articles/columns/column-articles/144-agile-coaching-in-british-telecom.

¹⁰ Bankwest goes Agile: project time slashed, Braue D (2010): zdnet.com/bankwest-goes-Agile-project-time-slashed-1339306091/.

¹¹ Suncorp goes Agile for 19k desktop integration project (2008): itnews.com.au/News/130927,suncorp-goes-Agile-for-19k-desktop-integration-project.aspx.

1: An Executive Brief on Agile

In order to fully appreciate the effectiveness of Agile methods, it is worthwhile taking a couple of minutes to understand the business environment that caused these approaches to be established in the first place.

A two-minute history of Agile

In the 1990s, the IT industry was plagued by the remarkably high failure rate of software development projects; projects that became notorious for their missed deadlines, substantially overrun budgets, faulty deliverables and dissatisfied customers. A handful of thought leaders in the industry believed that these IT project failures could be attributed to three key factors: over-planning, insufficient communication and ‘all-at-once’ delivery.

Over-planning

IT software projects traditionally began with the production of extensive ‘up-front’ documentation, including project plans, functional requirements, system design specifications, and technical architectural designs. These documents, which often took months to produce (and even longer to get approved), were intended to ensure that the developed software would align with user requirements. In reality, however, these documents only served to provide corporate managers with a false sense of security in the expenditure of their IT budgets; and to ensure that delivered software

¹² Is Agile Development Only for Nerds?, Matta E (2008): <http://radiowalker.wordpress.com/2008/10/07/is-Agile-development-only-for-nerds/>

1: An Executive Brief on Agile

would be substantially misaligned with the ongoing – and changing – needs of the business.

The biggest problems with organizations relying upon up-front documentation were:

- the lack of responsiveness to ongoing changes in user requirements, market demand, internal resource availability and the capabilities of the underlying technologies
- the tendency for stakeholders (e.g. business areas, customers) to:
 - not clearly articulate their requirements (which were then left to the discretion of the technical team to interpret)
 - want everything under the sun (resulting in highly critical business requirements being lost in a sea of extraneous requirements)
- the inevitable misalignment between text descriptions of the users' needs and the resulting software.

The bottom line is that software products delivered to meet these up-front design documents were destined to fail – and businesses were losing millions in the process.

Insufficient communication

The second overwhelming driver in the ongoing failure of software development projects in the 1990s was the traditional – and often deliberate – separation of the business areas that required the software and the technical staff responsible for delivering the solution (i.e. development in a vacuum).

1: An Executive Brief on Agile

Once the big up-front design documents for an IT project were finalized, they were generally handed over to the technical team for development. The technical team was then sent back to their desks (often located in a separate section, floor or even building from the business areas) with a pile of paper and an immutable deadline. The next time that the technical team interacted with the business area was when they installed the resulting software on the users' machines for acceptance testing (or, even worse, production use).

This isolation created inevitable issues with the resulting software, including:

- user requirements left to the interpretation of the technical team members, without the benefit of understanding the business context
- the inevitable disconnect between the two-dimensional concept proposed in the documentation and the manifestation of that concept into tangible screens that the user could interact with
- not allowing for changes to business requirements that may have occurred between the time that the user was last consulted and the months (and sometimes years) that followed before the resulting software was installed on their system.

All of these factors resulted in the delivery of software that was frequently misaligned to the needs of the business users, including inadequate workflows, system errors, critical design flaws and features that were rarely (or never) used by the business – with no remaining budget or resources available to address these issues.

1: An Executive Brief on Agile

‘All-at-once’ delivery

Software development projects in the 1990s depended heavily on ‘waterfall’ project management methodologies, where analysis, design, development, testing and delivery stages were undertaken serially, requiring the full completion of one activity before the next one could begin. The use of waterfall methodologies on these projects meant that software design could not begin until all of the requirements analysis was complete; software testing could not begin until software development was complete; and software was not delivered to the users until all of the preceding stages had been completed.

This use of waterfall approaches in the IT industry was intended to reduce business risk in project delivery, requiring each step to be completed to management’s satisfaction before further expenditure was incurred. In reality, waterfall approaches significantly increased the risk of IT project failure by:

- mandating big up-front documentation (with all of its related issues)
- discouraging responsiveness to changing requirements as the project evolved
- creating ‘silos’ of ownership that reduced communication across project team members.

Perhaps the most risky impact of these waterfall approaches, was delaying the delivery of tangible business outcomes until the very end of the project – when problems in the software are the most evident and changes to the software are the most costly.

1: An Executive Brief on Agile

Instead of enabling the organization to manage expenditures and risks throughout the software development project, executives were faced with an all-or-nothing proposition: keep pouring resources into a failing IT project, so that at least some value can be recovered from the previous investment, or end the project midstream and receive no tangible benefit to the organization. The ‘all-at-once’ delivery approach often left these executives with no other options.

There were, of course, other factors that influenced the high failure rate of software development projects in the 1990s, including limitations in technology and the lack of availability of skilled technical resources. However, the three issues outlined above – over-planning, insufficient communication and ‘all-at-once’ delivery – were factors that were *within the control of the organization to change*.

In order to combat the widespread failure of IT projects, a group of innovative thought leaders¹³ began to develop strategies and practices that were specifically designed to address these three issues. It is their insight, along with the contribution of many others that followed, which has resulted in the proven, business-value-driven Agile methods that are used throughout the world today.

¹³ Including Kent Beck, Martin Fowler, Alistair Cockburn, Jeff Sutherland and Ken Schwaber.

1: An Executive Brief on Agile

The 10 core business benefits of Agile

The business value that Agile methods provide to organizations can be distilled into the following *10 core Agile business benefits*:

1. **Ongoing risk management:** by regularly confirming and adjusting requirements based on ongoing interaction with stakeholders; and by delivering fully functional and tested software features, so that technical risks are identified and mitigated throughout the process.
2. **Ongoing control of budget expenditure:** by providing decision makers with the opportunity to review and assess the business value of deliverables in each iteration, with the option to adjust, postpone or stop ongoing funding based on the return on investment (ROI) of delivered work.
3. **Rapid delivery of tangible outcomes:** by focusing team efforts on regularly producing fully functional and tested software features.
4. **A focus on the highest-priority features:** by continually working with key stakeholders to confirm and adjust the work undertaken by the project team to align with the most current priorities of the organization.
5. **Strong alignment with business requirements:** by directly involving key stakeholders in the initial and ongoing review of developed software, and by incorporating their feedback throughout the process.
6. **Responsiveness to business change:** by adjusting work throughout the process to incorporate organizational, industry and technology changes.

1: An Executive Brief on Agile

7. **Transparency in status tracking:** by regularly providing tangible outputs for stakeholder review, combined with open communication channels and centrally available status-tracking tools.
8. **Substantially higher quality outputs:** by incorporating rigid testing regimes throughout the software delivery process; and by working regularly with stakeholders to confirm the usability and business value of delivered features.
9. **Greater employee retention:** by creating work environments that are based on high communication, empowering the team and trusting their expertise, encouraging innovation, and regularly acknowledging the team's contribution to the organization.¹⁴
10. **Minimized whole-of-life software costs:** by incorporating usability, quality and extensibility into software solutions throughout the delivery process, which reduces support and maintenance overheads; and by allowing additional functionality to be integrated into the solutions more cost effectively.

These combined benefits position organizations to receive a range of *real productivity gains*¹⁵ from their software development activities. They enable

¹⁴ From the team member's perspective, this collaborative and supportive work environment is arguably the most important business benefit of using Agile methods.

¹⁵ See www.realproductivitygains.com for further detail on identifying and quantifying real productivity gains.

1: An Executive Brief on Agile

these organizations to bring IT development work back under business control.

The following section provides an overview of some of the most common Agile methods, and identifies how each is designed to deliver these business benefits to your organization.

Common Agile methods

Figure 1, overleaf, summarizes the extent to which ten of the most common Agile methods deliver the *10 core Agile business benefits* identified in the previous section:

1: An Executive Brief on Agile

	Aligning Risk	Aligning Control of Budget	Rapid Delivery of Tangible Outcomes	Focus on Highest Priority Features	Strong Alignment With Business Requirements	Responsiveness to Business Change	Transparency in Project Tracking	Stakeholder Input	Quality Outputs	Greater Employee Satisfaction	Lower Software Costs
Scrum	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Dynamic Systems Development Method (DSDM)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Feature-Driven Development (FDD)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lean Development (Lean)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Extreme Programming (XP)	✓	Indirectly	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rational Unified Process® (RUP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Agile Unified Process (AUP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Kanban	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Scrumban	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Scaled Agile Framework® (SAFe®)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 1: Breakdown of the 10 core Agile business benefits by Agile method

1: An Executive Brief on Agile

Each of these methods is described in further detail in the following sections.

Scrum

Scrum is an iterative project management method most commonly used for Agile software development projects, but suitable for any project-based work. Scrum provides a framework for business areas to identify and prioritize work required, and for project teams to commit to the subset of priority items that they believe can be delivered in each two- to four-week iteration (or 'sprint').

Scrum requires the nomination of resources to provide key roles in the project, including:

Scrum is an Agile method for project management that involves:

- delivering software in time-boxed iterations
- focusing on the highest business-value software features in each iteration
- interacting directly with business users to confirm ongoing software usability, relevance and business value throughout the process.

- the *Product Owner* who represents the needs of the business, and is responsible for documenting and prioritizing solution requirements as input into ongoing planning
- the *Scrum Team*, a cross-disciplinary team charged with undertaking the agreed work in each sprint

1: An Executive Brief on Agile

- the *ScrumMaster* who facilitates the team's work, removing project impediments and ensuring that appropriate Scrum practices are being followed by the team.

Core to the success of Scrum are two activities that are undertaken for each iterative sprint:

- the *Sprint Planning Meeting*, held at the beginning of each sprint, where the Product Owner, ScrumMaster and Scrum Team review the highest-priority items identified by the Product Owner and agree on the subset of priority items that will be included in the forthcoming sprint
- the *Sprint Review*, at the end of each sprint, which includes a demonstration of work completed in that sprint and a retrospective review of the work undertaken to enable continuous improvement for subsequent iterations.

Scrum also encourages teams to engage in *daily stand-up meetings*, short update sessions held each morning that enable the team to quickly review their completed and planned work and address any hurdles.

The progress of the Scrum Team's work is communicated to stakeholders through monitoring and measurement tools, such as:

- the *executive dashboard*, a report that summarizes the work within (and across) Agile teams for easy progress monitoring across the organization
- the *product backlog*, a reporting tool that enables both stakeholders and project teams to

1: An Executive Brief on Agile

monitor the progress of work against the agreed business requirements¹⁶

- the *sprint backlog*, a reporting tool that enables teams to monitor and manage their actual day-to-day work.

Scrum is arguably the most commonly used Agile approach worldwide, and a range of professional courses are available to certify people for roles such as the ScrumMaster and Product Owner. These courses foster the consistent and effective application of these roles in every organization. They are also intended to address common areas of misapplication in Scrum implementations, such as the mistaken view of the ScrumMaster as a project manager (instead of a facilitator).

This collaborative approach to project management and status tracking – combined with regular reviews of tangible outcomes and business-driven prioritization of ongoing work – positions Scrum to deliver all of the *10 core business benefits of Agile*.

¹⁶ Some Scrum teams also use a Release Backlog to identify and manage subsets of capabilities that are planned for delivery in scheduled releases.

1: An Executive Brief on Agile

Dynamic Systems Development Method

The Dynamic Systems Development Method (DSDM) framework is another iterative approach to managing Agile software development projects. It has its roots in Rapid Application Development (RAD), resulting in a strong emphasis on building prototypes and confirming the feasibility of the solution with the business prior to undertaking full development activities. This is evidenced by DSDM requiring Stakeholder Workshops, a Feasibility Report, a Feasibility Prototype and a Business Study to be undertaken prior to full implementation.

The practices that underpin DSDM are at the very heart of Agile methods, including active user involvement throughout the process, iterative and incremental development, frequent delivery of tangible outputs and empowering the team. Ongoing testing and quality control of software throughout the process are also emphasized. These practices combine to enable DSDM, like Scrum, to

Dynamic Systems Development Method (DSDM)

is an Agile method for project delivery that involves:

- delivering software in time-boxed iterations
- prototyping and documenting the software solution prior to undertaking full development activities
- collaborating with users, producing tangible outputs, and ensuring quality management throughout the process.

1: An Executive Brief on Agile

deliver all of the 10 core business benefits of Agile.

Unlike Scrum, however, the DSDM framework requires a range of artifacts (e.g. development plans, functional models) to be developed at each phase of the project to provide ongoing confirmation that planned work is aligned with the needs of the business.

Although the approaches differ, both Scrum and DSDM have the same core objective – the delivery of high business-value outcomes in controlled, iterative timeframes. Scrum provides a high-level framework for achieving this objective, and relies on communication between the participants to ensure that work undertaken meets ongoing business needs. DSDM provides a more structured framework to achieve this objective, requiring proposed work to be documented and confirmed prior to continuing to the next stage.

Copyright IT Governance Ltd

1: An Executive Brief on Agile

Feature-Driven Development™

Feature-Driven Development (FDD™) is an Agile method that combines elements of iterative project management with practices that are specific to software development. The basic driver of FDD™ is providing incremental value to the business by delivering complete, working product capabilities (i.e. software ‘features’ and ‘feature sets’) in every iteration.

FDD™ requires close collaboration with the business areas to establish an up-front ‘domain model’ of the business problem that the proposed system is intended to address. Once this domain model is identified, it is broken down (decomposed) into smaller units (i.e. ‘features’) that are able to be developed and delivered iteratively. Team members are then assigned to build nominated features which, once successfully tested, are incorporated into the larger system.

Feature-Driven Development (FDD™) is an Agile method that combines iterative project delivery with software development practices by:

- having teams model the business problem up front
- decomposing the model into smaller features and feature sets
- integrating selected feature sets into the overall software solution through iterative releases
- keeping a strong focus on collaboration with users, production of tangible outputs, and quality management throughout the process.

1: An Executive Brief on Agile

In many respects, FDD™ works from the same underlying principles as other Agile methods (e.g. Scrum), in that the project team works closely with the business areas to deliver regular, incremental value to the organization. However, FDD™ is far more prescriptive about defining the boundaries of the solution up front, assigning specific roles and responsibilities to the team members, and controlling the scope of each team member's work during the actual software development process.

Based on the above, there is an argument to say that, although FDD™ delivers all *10 core business benefits of Agile*, its ability to deliver 'responsiveness to business change' is limited. This is because the structure of FDD™ makes it less responsive to ongoing organizational, industry or technology changes that do not fit in with the originally identified domain model.

Lean development

Lean development (Lean) is an Agile method that combines elements of iterative project management with best practices in software design and development. Lean stems from the Lean manufacturing processes that originated as early as 1922¹⁷, but most closely aligns with the principles

¹⁷ *My Life and Work*, Ford H with Crowther S, Garden City Publishing Company, Inc. (1922), ISBN 9781406500189.