



Valuing Agile
The Financial Management of Agile Projects
ALAN MORAN



Valuing Agile

The Financial Management of Agile Projects

London: TSO



Published by TSO (The Stationery Office), part of Williams Lea, and available from:

Online

www.internationalbestpractice.com

www.tsoshop.co.uk

Mail, Telephone, Fax & E-mail

TSO

PO Box 29, Norwich, NR3 1GN

Telephone orders/General enquiries: 0333 202 5070

Fax orders: 0333 202 5080

E-mail: customer.services@tso.co.uk

Textphone 0333 202 5077

TSO@Blackwell and other Accredited Agents

International Best Practice (IBP) is a framework-neutral, independent imprint of TSO. We source, create and publish guidance which improves business processes and efficiency. We deliver an ever-expanding range of best-practice guidance and frameworks to a global audience.

© The Stationery Office 2016

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the permission of the publisher.

Applications for reproduction should be made in writing to The Stationery Office Limited, St Crispins, Duke Street, Norwich NR3 1PD.

The information contained in this publication is believed to be correct at the time of manufacture. Whilst care has been taken to ensure that the information is accurate, the publisher can accept no responsibility for any errors or omissions or for changes to the details given.

ITIL® is a registered trade mark of AXELOS Limited.

PRINCE2® is a registered trade mark of AXELOS Limited.

PRINCE2 Agile® is a registered trade mark of AXELOS Limited.

RESILIA™ is a trade mark of AXELOS Limited.

A CIP catalogue record for this book is available from the British Library.

A Library of Congress CIP catalogue record has been applied for.

First published 2016

ISBN 9780117082885

P002776710 03/16

Contents

List of figures	v
List of tables	vi
About this guide	vii
Foreword	ix
Preface	x
Acknowledgements	xi
1 Introduction to Agile	1
1.1 The Agile approach	1
1.2 Agile software development	3
1.3 Agile techniques and practices	5
1.4 Product development	6
1.5 Project management	7
1.6 Related disciplines	9
1.7 Financial challenges	10
1.8 Concluding remarks	12
2 Requirements, estimation and contracting	13
2.1 Overview	13
2.2 Requirements	15
2.3 Estimation	17
2.4 Team velocity	19
2.5 Prioritized incremental delivery	20
2.6 Types of contracts	22
2.7 Agile contracting	23
2.8 Government contracts	25
2.9 Concluding remarks	26
3 Strategy, budgeting and expenditure	27
3.1 Strategy, portfolios and projects	27
3.2 Agile budgeting	28
3.3 Financial statements	32
3.4 Capital and operational expenditure	34
3.5 Concluding remarks	37
4 Profitability and Agile financial performance	39
4.1 Understanding profit	39
4.2 The significance of timing	41
4.3 Agile impact on ROI	43
4.4 Worked example	46
4.5 Concluding remarks	51

5	Value and benefits realization	53
5.1	Understanding value	53
5.2	Social value	55
5.3	Communicating value	56
5.4	Earned value management	58
5.5	Benefits realization and assessment	61
5.6	Concluding remarks	62
6	Managing uncertainty	63
6.1	Understanding risk	63
6.2	Agile risk management	64
6.3	Costing risk	68
6.4	Sensitivity analysis	69
6.5	Foreign exchange risk	72
6.6	Credit risk	75
6.7	Decision point analysis	76
6.8	Concluding remarks	78
	Appendix A Analytical treatment of cash flows	79
	Glossary	81
	Bibliography	91
	Index	95

List of figures

Figure 1.1	The Scrum process model	7
Figure 1.2	The DSDM process model	8
Figure 2.1	The relationship between requirements, estimation, transaction costs and governance	14
Figure 2.2	User story map	17
Figure 2.3	The cone of uncertainty	18
Figure 2.4	Planning Poker cards	18
Figure 2.5	Project variables (traditional versus DSDM)	23
Figure 3.1	Agile cycle of planning, appraisal, funding and validation	30
Figure 3.2	DSDM partition of capital and operational expenditure	36
Figure 4.1	Computation of NPV using Microsoft Excel	42
Figure 4.2	Comparison of NPV profiles for traditional (dashed) and Agile (solid) projects	44
Figure 4.3	Decision point analysis of Agile projects	45
Figure 4.4	Discounted Agile cash flows	48
Figure 5.1	Example of a burn-up chart showing schedule variance during an iteration	60
Figure 6.1	Mapping of risk exposure (including ranges) to risk response strategies	66
Figure 6.2	Illustrative risk burn-down chart	67
Figure 6.3	Hedging against foreign currency risk	74

List of tables

Table 1.1	Some common Agile techniques	5
Table 1.2	Some common Agile practices	6
Table 1.3	Five levels of Agile planning	10
Table 4.1	Project direct and indirect costs (i.e. overheads)	40
Table 4.2	Comparison on basis of apportionment	40
Table 4.3	Discounted cash flows and net present value	42
Table 4.4	Anticipated project costs and revenues	46
Table 4.5	Discounted cash flows for Agile and phased project approaches	47
Table 4.6	Summary of financial performance metrics	50
Table 5.1	Project tracking metrics	60
Table 6.1	Scenario-based project and operational cash flows	70
Table 6.2	Sensitivity analysis of project cash flows	71
Table 6.3	Transaction costs and exchange rates	74
Table 6.4	Strategies for managing exchange rate risk	75

About this guide

Who's it for?

This publication addresses the needs of both Agile practitioners and financial managers. It is suitable for business sponsors and executives (including product owners), Agile project managers, Scrum masters, team leaders, project management office staff and anyone else with an interest in the financial impact of Agile projects. By reading and understanding the chapters that follow, you will be better able to articulate the benefits of Agile in clear and uncomplicated language.

Why do you need it?

With the rise of interest in Agile projects, their financial management is becoming increasingly important. If you are responsible for the success of your Agile projects, then you will need to have a solid understanding of how to manage your finances! This book provides practical guidance on Agile requirements and financial and risk management, covering topics such as Agile contracting and budgeting while showing you how to cope with the impact of change and uncertainty on your project finances.

Who's it by?

Alan Moran



Alan Moran is an Agile thought leader and IT management professional with extensive senior-level experience of implementing and managing high-value IT and business solutions in public and private-sector organizations. An MBA graduate of the University of Strathclyde, he is also a doctorate holder in mathematics and has earned numerous certifications in Agile, project and risk management, IT service management and related disciplines. He is the managing director of the Institute for Agile Risk Management (IARM), a Swiss-based institution that exists to promote the principles and practices of Agile management within the context of the Agile enterprise. He is also the author of several publications, including *Agile Risk Management* (2014), *Managing Agile: Strategy, Implementation, Organisation and People* (2015) and *Agile Risk Management and DSDM* (2015). Alan has maintained a keen interest in establishing and optimizing Agile processes throughout his career, and is a regular speaker at international conferences on topics relating to Agile management.

How is it organized?

Each chapter is self-contained and can be read independently. Worked examples illustrate clearly how Agile financial principles can be applied to your daily work. A summary of what is covered in each chapter is as follows:

- **Chapter 1** Provides an overview of Agile principles and practices and introduces two specific methodologies that focus on product development (Scrum) and project management (DSDM). The chapter concludes by positioning Agile within the wider family of related disciplines that includes Design Thinking, Lean Startup and Beyond Budgeting.
- **Chapter 2** The Agile process is explored in greater detail by looking at how requirements are solicited and effort is estimated. In this context the contractual aspects of supplier management in both the private and public sectors are examined.
- **Chapter 3** This takes a look at the wider strategic planning and budgeting processes within organizations. It considers how Agile budgeting ought to be performed and how capital and operational cost classifications are derived.
- **Chapter 4** Examines what is meant by profit and identifies the key drivers of value in Agile projects. These are then illustrated by means of an extensive worked example.
- **Chapter 5** Describes how notions of value are captured in Agile projects and how to measure their benefits using tools such as Agile Earned Value Management.
- **Chapter 6** Presents a detailed overview of financial risk by describing how to cope with uncertainty during appraisal analysis. The chapter concludes with a discussion of a number of sources of risk such as cost, credit and foreign exchange, and the application of real options to Agile projects.

Foreword

Financial planning and risk management for Agile initiatives are topics of increasing interest. A book addressing these areas has been needed for some time and this publication provides a deep dive into these highly relevant topics. Readers will gain much value and understanding from the detail presented here. The theory and application of financial and risk management practices, as well as guidance to support their alignment to an Agile approach, provide a fascinating read.

Steve Messenger

Chair, DSDM

Preface

A central tenet of the Agile philosophy is its attitude towards change; it requires an adaptive approach to planning. Agile takes a positive stance in relation to emergent opportunities and there is a balance between the reward and risk incurred in pursuing them. Agile has to embrace not only requirements (e.g. user stories) and an understanding of their value from the customer's perspective, but also a means of funding them that is flexible enough to respond to material changes in the wider environment. Indeed, the creation of this publication was itself the result of an Agile publishing collaboration between TSO and the Institute for Agile Risk Management. Working closely with a community of reviewers, we revised each chapter over several iterations. At the end of each we asked for feedback, which we integrated into the next version. In a spirit of openness that is commonly found in Agile projects, we were able to engage in discussions that drew upon the strengths and diversity of perspectives found among the reviewers.

Valuing Agile: The Financial Management of Agile Projects provides a framework for the financial management of Agile projects built around the notion of Agile budgeting. In Agile budgeting, planning is linked to appraisal, from which we can derive incremental funding requirements. These requirements are then used to develop and deliver a solution that tests the underlying business assumptions of the project. This process makes it possible to grasp how the combination of the prioritization of 'user stories' by value, along with their iterative development and incremental delivery, reshapes the cash-flow structure of a project. This has a positive influence on net profit as measured by any of the commonly used financial metrics. In order to recognize this fundamental dynamic of Agile project finances, we need to understand what constitutes profit, how costs are treated and how expenditure is classified within the organization. We also need to understand how the value created by Agile projects is reported through practices such as incremental discounting. This provides the necessary foundations for financial risk management of Agile projects through the use of sensitivity analysis and real options. In this way relevant information, which is incorporated as it comes to light, is used to determine the future direction of the project.

Using practical illustrations and examples from common Agile methodologies, *Valuing Agile: The Financial Management of Agile Projects* explains how to adapt traditional financial management techniques for Agile projects, enabling practitioners to apply them in their daily work. It explains the core concepts clearly, and the links between product development, Agile contracting, budgeting and project appraisal, value realization and assessment and financial risk management are presented in a concise and coherent manner. This ensures that many of the common misunderstandings and pitfalls are avoided. By addressing the needs of both Agile practitioners and financial managers, this publication establishes a common ground on which it is able to explain the management of Agile project finances.

Acknowledgements

This publication is based on direct experience and exchanges within the wider community of Agile practitioners and project managers. Their wisdom and insights have added perspective and meaning to this work. To those who have contributed by whatever means, my sincerest gratitude is owed. I am particularly grateful to the reviewers listed below; their generous remarks and comments were much appreciated.

About the reviewers

Gopi K. Aitham

A lifetime learner and entrepreneur at heart, Gopi provides training and coaching to help individuals and companies deliver projects more effectively. Certified as a SCRUMstudy trainer, he has advised more than 1,000 professionals on Scrum, Agile, Kanban, PMP certification, Microsoft Project and Primavera P6. He is managing director of Mahtia.com and founder of brands 24x7Coach.com and KidVentures.in. He is an active blogger and reviewer of project management books.

Ştefan Bârgăoanu

An entrepreneur and a pioneer of the local Agile movement in his home town of Iasi, Romania, Ştefan has been helping software development teams to improve since the mid-2000s. He is particularly interested in the connections between Lean and Agile and how Agile can be applied outside the realm of software development. As an Agile coach, Ştefan believes that help is good wherever you can get it, so he tries to take what is useful from all of the methodologies, giving credit to them where it is due.

Sekhar Burra

Sekhar is an enterprise transformation coach and a change management consultant focused on changing the world of work. He has been awarded several globally recognized accreditations, including those of certified Scrum coach, Scaled Agile Framework (SAFe®) programme consultant, PMP and Management 3.0 practitioner. He is a lively speaker, Agile enthusiast, blogger and avid learner. He is an engineering graduate and has an MBA from a top business school in India. He is affiliated to several user groups and Agile communities in India.

Laura Hind

Laura Hind grew up in Honolulu, but now lives in northeast Ohio, where she oversees the project management office for MTD Products Inc. Currently she is part of a team that is implementing a scaled Agile transformation to the IT department to improve the intake

process for IT work as well as the requirements communication between the business and development groups. Her team of project managers/Scrum masters promote the Agile techniques and practices for Scrum, Kanban and Lean to facilitate the movement of stories through the development process. She supervises the Scrum masters of four large teams and plans to roll out the process to the remaining resources in IT.

Alan S. Koch

Author, speaker and consultant, Alan is the founder of ASK Process Inc. and draws on 15 years of experience in IT consulting and training. He authored the book *Agile Software Development: Evaluating the Methods for your Organization* (2005) and has peer-reviewed articles in various software journals. He presented the keynote speech 'What is Agile? And why should testers care?' in April 2009 at the Tianjin International Conference on Software Testing, and has performed numerous Agile presentations at other conferences.

Greg Lucas-Smith

Greg Lucas-Smith is the director of Resourceful Software, a software development and Agile coaching company based in Perth, Australia. He specializes in the transformation of dysfunctional teams into highly efficient and responsive client collaborators. His many years of software development using a broad range of tools, techniques and methodologies have given him a deep understanding of the problems associated with software development. His recent interest in applying Agile techniques to business management problems, encouraging stronger communication between business hierarchies, has produced great results.

Miki Matsuoka

Miki Matsuoka is project manager of software and research and development for both Waterfall and Agile processes. Miki has led international and cross-cultural projects involving companies, academic institutions and government. These projects have focused on maximizing engineers' competence and increasing the value of projects. Miki currently coaches people in Agile methodology.

Kathi Paquet

Kathi Paquet PhD is an IT professional with more than 20 years' experience, including portfolio and project management, forecasting, budgeting and risk management. She is currently a senior manager for Centric Consulting and leads various Agile transformation and SAFe® implementations. She is a committee member for the St Louis Agile LINC and Agile executive council, and has also developed a variety of Agile courses and communication strategies for executive councils to help with Agile transitions.

Carlos Sánchez-Sicilia

Carlos is a consultant and managing director of the Panama-based company Talentia, which provides a balanced portfolio of high-quality consulting and training services in enterprise architecture, project management, IT governance, business strategy and entrepreneurship. A computer science graduate, he also holds master's degrees in business administration and financial management.

Scott Tunn

Scott is a consultant, trainer, author and managing director of Adelaide Consulting Group. He works with a wide variety of clients in many countries, using innovative ideas from IT service management and business management to create and increase value for his clients and their customers. He holds degrees in business management and administration, and is a certified expert in ITIL®, as well as holding RESILIA™, PRINCE2® and ISO 20000 certifications. He contributes to the global community via Standards Australia and shares his expertise widely, regularly presenting at events and contributing to books, white papers, blogs and guides on all aspects of IT.

Darren Wilmshurst

Darren is a director of Radtac Ltd, a global Agile transformation company based in London. With 16 years' experience in retail banking, he is an associate of the Chartered Institute of Bankers and senior personal banking manager within the Guildford group of branches. Darren is also a practitioner for PRINCE2, PRINCE2 Agile®, DSDM Atern Agile PM, Kanban and APMG, a certified Scrum master, and a SAFe programme consultant. He is a co-author of the British Computer Society book, *Agile Foundations: Principles, Practices and Frameworks* (2015).

1 Introduction to Agile

As set out in the *Manifesto for Agile Software Development* in 2001, the Agile philosophy prescribes an approach to solution development and project management that accepts change and responds to the needs of customers. It belongs to a wider movement (including Lean, Design Thinking and Beyond Budgeting) that shares these values together with a desire to deliver value amidst uncertainty. Exploring the Agile approach leads to a better understanding of how financial practices founded on command-and-control principles (e.g. budgetary controls) are actually inhibiting innovation. Accordingly, there is a need for an Agile financial approach to project appraisal that includes a more holistic and adaptive performance assessment perspective.

1.1 The Agile approach

Agile is a term that describes a set of values and principles expressed as preferences that are based on group learning and dynamics fuelled by short feedback cycles. These cycles encourage an emergent solution to be delivered incrementally in a collaborative and responsive manner. These are expressed in terms of the following four preferences (that acknowledge the value of the items on the right, whilst valuing those on the left more):

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan.

These four items, together with the following 12 principles, constitute the Agile Manifesto:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.

- Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity – the art of maximizing the amount of work not done – is essential.
- The best architectures, requirements and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Agile solution development recognizes the inherent limitations in plan-driven, phased approaches (sometimes referred to as 'Waterfall'), relying on estimates that may be questionable in uncertain environments. It does so by embracing change and promoting iterative development and incremental delivery; it is a culture of continual feedback and learning based on trust and nurtured by self-organizing teams. Thus the paradigm of the Agile organization is a small, cohesive and diverse team of individuals consisting of specializing generalists (i.e. detailed expertise in one or more core disciplines complemented by broader skills in a range of other fields) who have the freedom within broadly defined parameters to shape their own practices. While there may still be a project manager or team leader, this role tends to be more facilitative and supportive; a style of leadership often referred to as 'servant-leadership'. Thus the essence of agility is to have strategies to cope with changes that arise from dynamic market environments, developments in technology or other sources of uncertainty.



Agile teams

Agile teams are usually small, consisting of between three and nine members, though figures vary by methodology. Individuals within the team may have a broad range of skills (e.g. a software developer with database and system administration skills or a business analyst who also has interests in graphical user interface and graphic design) that enable them to share tasks more easily, work together and learn from each other. It is generally considered best practice to have the entire team in the same location, where open, shared conversations can contribute to wider dialogue within the team; this is known as 'osmotic communication'. This atmosphere of openness within Agile environments also encourages cooperative and supportive group dynamics when an individual seeks help (as evidenced when team members 'swarm' to assist a colleague) or it can result in social sanction from the group if an individual fails to pull their weight.

Being Agile can be disruptive, owing to the way it turns many things on their heads within an organization. For example, the boundaries between functional disciplines (e.g. requirements engineering, software development, testing and quality) often become blurred as do the organizational units to which they belong, as teams become more mixed and cross-functional. These features of Agile working practices challenge self-identity (e.g. what is meant by the roles of manager or technician?) as indispensable specialists become increasingly rare and social skills (e.g. negotiation and conflict resolution) are valued at least as much as technical ones. In this environment, what counts is the ability to engage in

organizational learning and sharing; this, in turn, relies on openness and flexibility on the part of, not only the team members, but also the organization. Team accountability also means that individual bonus schemes (known to be counterproductive in most organizations) are replaced with group-based rewards. Needless to say, embracing Agile extends beyond the mere adoption of a methodology and reaches to the core values and identity of the enterprise.



Salesforce.com

In 2006 Salesforce.com undertook a remarkable enterprise-wide transformation that saw it switch across to Agile practices over a three-month period (followed by a further year of continual improvements). The results were dramatic: there was a 500% increase in value delivered to its customers compared with the previous year. This was achieved through a 94% increase in released features as staff reported higher levels of transparency and productivity. Indeed, despite having initial misgivings about the transformation programmes, teams expressed high levels of satisfaction both with the new working practices and with the quality of products and services they delivered. Meanwhile, customers, pleased with the timeliness of product releases and upgrades, were more likely to recommend Salesforce.com to their colleagues. The success of this programme relied to a large degree on the use of cross-functional teams, a return to the core values of the organization and a willingness to engage with customers throughout the solution development process.

1.2 Agile software development

Already in widespread use in the manufacturing industry, Agile practices evolved in the IT sector during the 1990s before finding expression in the Agile Manifesto. Today Agile is a mainstream practice for software development in the IT sector. The traditional approach in the IT sector had been based on the Software Development Lifecycle (SDLC), interpreted as a sequence of phases: requirements gathering (during which specifications are created); analysis and design; solution implementation and testing; culminating in the release of a software package. In fact, rather like a relay race in which each leg must be completed before the baton is passed onto the next runner, each phase of the SDLC is expected, according to the traditional interpretation, to be finished before work on the next can commence. The belief prevailed that it was necessary and possible to plan in advance all activities in detail and that while uncertainty might exist, this could be remedied by sufficiently detailed analysis, based on a presumption of clearly expressed needs of the customer. This promoted the notion that specialists (such as business analysts, software developers and testers) should be involved in each phase and that such individuals must be coordinated by a project manager, with the role of customer being limited to clarifying requirements and performing acceptance testing of the final product.

So where might problems lurk with this approach? The reality is that software development often involves a journey into the unknown, and may be further buffeted by market competition or shifts in corporate strategy or vision. Not only is it unrealistic to expect customers to be able to state precisely what they want without changing their minds; team specialists may also be working with new technologies for which reliable estimates (required

by traditional planning approaches) cannot be made. In fact, there are numerous examples today of disruptive new technologies (i.e. those upsetting the status quo in the marketplace) that addressed needs which at the time were hard to capture or express clearly until users were able to meaningfully interact with the emergent products or services. Examples include Twitter, Netflix, Spotify, Instagram, Facebook and LinkedIn. It therefore makes more sense to break down the development process into smaller iterations and integrate the customer more often. For example, the customer may assist in the planning for each iteration; they may be on-call should issues or the need for clarification arise during development, and they may be present at the end of the iteration to review what has been achieved. To understand the customer's needs it may be necessary to build prototypes or to experiment with designs. In this way, direct experiences and deeper insights are gained through collaboration with the customer. Seen as an exercise in organizational learning, the Agile process takes on entirely different nuances. Agile therefore requires generalists rather than specialists, who need to be self-organized and empowered, rather than taking a command-and-control approach. The result is a way of working that permits an emergent solution to grow, for it to be continually validated and ultimately satisfy the needs of the customer.

The traditional engineering approach to product development in IT, based on the phased interpretation of the SDLC, also fails in a number of other ways beyond the time it takes to get the right product to market. From a psychological viewpoint, capable individuals in a team who are denied autonomy or accountability often lack the level of commitment found on an Agile project that grows from identification with its objectives and a sense of purpose in achieving them. This is because if an individual is directed in a command-and-control environment then their motivation is limited to achieving the outcomes expected of them as defined narrowly in terms of their respective specialisms (i.e. they can hardly be blamed for doing what they were told to). This also means that such individuals will often frame productivity in terms of their outputs (e.g. amount of code produced) rather than the learning that takes place when they engage with others. For example, pair programming, an Agile technique in which two developers share a terminal to collaborate on a problem, is often misunderstood as an inefficient means of working. However, studies have shown that when working in pairs, individuals are considerably more likely to do the right thing (e.g. write unit tests or refactor the code where necessary) which saves time in the long run; this is referred to as reducing technical debt. Pair programming is therefore all about learning and avoiding wasted effort, rather than trying to accomplish the task at hand twice as fast.



Zara

What might an Agile environment look like outside the IT sector? Inditex, the clothing manufacturer and retailer best known for its Zara brand, is a well-known proponent of Agile management practices. They enable it to deliver new designs to the shop floor in an astonishing three working weeks. It does so by integrating design and manufacturing teams in an effort to reduce communication barriers, enforcing timeboxed delivery of small batches, and ensuring flexibility by leaving space in its retail stores in order to respond to unanticipated demand. Teams at Zara are self-organized and work to short cycles driven by customer feedback. This approach enables Zara to exceed its industry peers in terms of sales, operating profits and stability.

1.3 Agile techniques and practices

Pair programming is just one of many techniques commonly employed by Agile teams (see Table 1.1). Working in a diverse collaborative team has been shown to broaden the experience and perspectives of all participants, enabling them to see more creative solutions to problems. Individual and organizational flexibility, when combined with such creativity, are essential elements of innovation and help teams to deliver new products and services that might not otherwise have been possible. So being Agile can be said to promote individual growth, as well as acting as a driver of innovation.

Table 1.1 Some common Agile techniques

Agile technique	Description
Continuous integration	Integration of changes into the product baseline facilitated by automated verification of the functionality and deliverability of the solution that establishes an early feedback loop
Nightly builds	Automated build and deployment of everything that has been achieved during the day into an environment that enables it to be used for purposes of assessment and testing
Test-driven development	The practice of creating tests prior to the implementation of code in order to improve validation feedback
Refactoring	The internal reorganization and improvement of a solution while preserving its external behaviour
Pair programming	The discipline of working alternately in pairs on a solution (with the purpose of avoiding wasted effort)

Agile employs a wide variety of practices, some of which are found in Table 1.2. Together with new organizational paradigms (e.g. self-organization and the use of generalizing specialists), these ensure that an Agile team possesses the necessary set of skills and social competencies in order to deliver value to the customer. Seen from this perspective, Agile behaviour can be considered as principally a cultural stance on solution development rather than a mere collection of values and principles.

In the period leading up to 2001 when the Agile Manifesto was signed (and since), there has been a plethora of different Agile methodologies. While these share the same broad values and principles and borrow extensively from each other, they have all found their own communities and unique cultures. They include approaches that emphasize software engineering techniques (e.g. Extreme Programming (XP) and Pragmatic Programming and Feature-Driven Development); product development (e.g. Scrum); project or portfolio management (e.g. DSDM, SAFe); other facets such as flexibility and decision frameworks (e.g. DAD, Crystal); or Lean thinking (e.g. Lean Software Development). While some methodologies have cross-fertilized (e.g. Scrum and XP), others have been developed over the course of several formal versions (e.g. DSDM). Two specific methodologies, one

primarily concerning product development (Scrum) and the other more focused on project management (DSDM), will be discussed further in sections 1.4 and 1.5 respectively and used throughout this publication to illustrate examples of financial practices. Though each has its own domain, this does not preclude the presence of the other. It is perfectly legitimate, for example, to refer to Scrum in a project setting or to a DSDM product development effort.

Table 1.2 Some common Agile practices

Agile practice	Description
Iteration planning	Planning session at the start of each iteration, often including the customer or their representative (e.g. product owner)
Daily stand-up	Short, structured daily meetings to review progress and plan events for the day
Review	Demonstration of the (partially) complete solution at the end of the iteration
Retrospective	Communal reviews of the performance of the underlying Agile process

1.4 Product development

Scrum, a product development methodology originating in the mid-1990s, has become (alongside its closest variants) the most popular Agile methodology today. Scrum owes its name to the sporting metaphor of collaborative effort in rugby that contrasts with the ‘relay race’ of traditional phased approaches. Work is performed in a series of iterations, known as sprints, that are timeboxed into periods usually lasting two to four weeks (see Figure 1.1). Though the product is potentially shippable at the end of each sprint, it is usually released in line with the customer’s schedule. Scrum is based on a model in which everything that might be needed in a product is recorded (in descending order of priority) in a product backlog to which new items may be added continually. At the start of each sprint a selection of items are taken from the top of the product backlog (i.e. highest-priority items first). The customer then helps to analyse them and make a detailed estimate to create the sprint backlog. From this a sprint goal statement is formulated that captures the spirit and intent of what is to be achieved by the end of the sprint. Sprint backlog items are then implemented and tested during the sprint with the aim of producing a potentially shippable product. At the end of the sprint the team demonstrates the results and reviews them with the customer. It then decides whether to release the product. Should the team decide not to release, the efforts are not wasted since the outputs will be carried over to the next sprint and will become part of a future release. The sprint concludes with a retrospective to discuss the interactions between people and any possible process improvements.

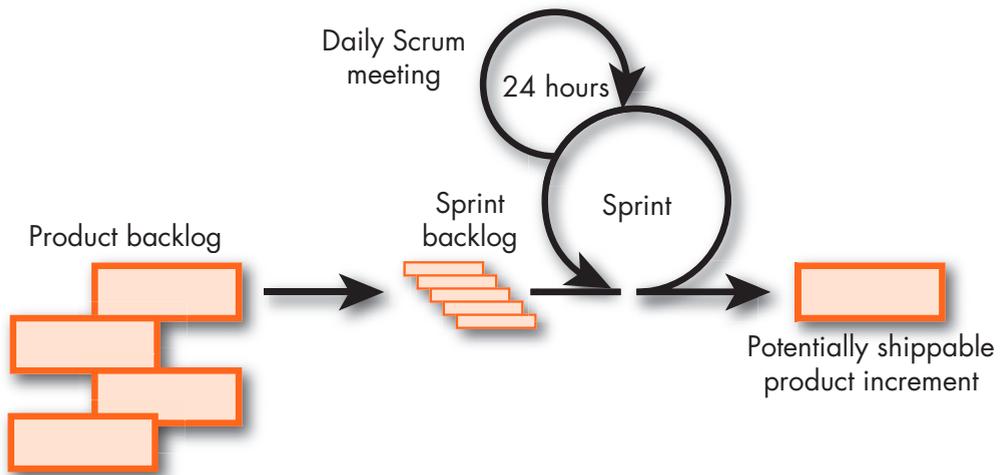


Figure 1.1 The Scrum process model

Overall responsibility for the product backlog lies with the product owner who must also ensure that the team understands its individual items. The product owner is supported by the Scrum master who ensures that all the participants understand the Scrum process and adhere to it. The only other role is the development team whose function is to analyse, implement and test the emergent solution. Scrum is limited to product development and does not consider what happens to the solution after it has been deployed other than the handling of requests for new features or fixing of bugs found in deliverables. The core elements of Scrum can be found in many other methodologies (e.g. SAFe, LeSS and DAD) that attempt to scale it up to multiple concurrent teams.

1.5 Project management

Though Agile practices are commonly used in product development, they are also project, risk and programme management techniques and can be used as part of the overall approach to strategic management within the enterprise. Dynamic Systems Development Method (DSDM), arguably the first truly Agile methodology, is an example of an approach that is firmly based on project management principles. Its core model (see Figure 1.2) is based on a phased approach, consisting of pre-project, feasibility, foundations, evolutionary development, deployment and post-project phases. During the feasibility and foundations phases, the basis for evolutionary development is established, from which solution increments are delivered via the deployment phase. All this activity is sandwiched between the pre- and post-project phases. DSDM refers to the iterations as timeboxes and uses a flexible prioritization and contingency-based approach to manage items from its prioritized requirements list that are worked out in detail in their respective timebox plans.

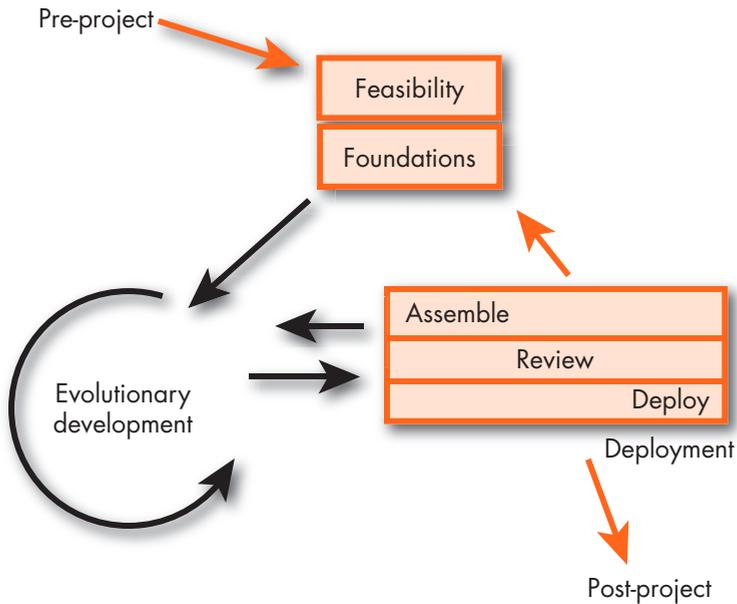


Figure 1.2 The DSDM process model

© Dynamic Systems Development Method Ltd. Reproduced by kind permission

DSDM has a fine-grained set of roles that includes a project manager, business representatives (business sponsor, visionary, ambassador, analyst and adviser) and solution development individuals (i.e. team leader, technical coordinator, solution developer, tester and adviser). It advocates specializing generalists (whom it refers to as T-shaped skilled individuals) whose work is governed by a set of eight guiding principles (e.g. focus on business need and quality; on-time iterative and incremental delivery; communication and collaboration, demonstration of control).

It is supported by a set of core practices such as facilitation, timeboxing or MoSCoW (see section 2.5), as well as a comprehensive set of planning, delivery and governance products (e.g. the business case, prioritized requirements list, timebox plan and benefits assessment).

DSDM is a comprehensive framework that caters for governance, quality and risk and can be scaled to manage Agile programmes capable of coping with a mixture of both Agile and non-Agile approaches. Owing to its management appeal, DSDM tends to be well suited to mature and regulated environments that might otherwise not have considered an Agile approach. It has been successfully integrated with a variety of traditional approaches (e.g. CMMI®, ITSM/ITIL and PRINCE2) and it is considered reasonable to embed other Agile approaches, such as Scrum, within its evolutionary development phase, thereby enabling it to combine both project and product management facets into a single framework.

1.6 Related disciplines

As the Agile movement evolves beyond IT, it is increasingly being used in concert with other innovative and design-driven approaches, with which it has found a particular resonance owing to shared values and perspectives concerning change and the management of uncertainty. In such situations Agile practices typically partner as the product development or project management function. The following disciplines are most commonly found in conjunction with the Agile approach:

- **Design Thinking** A framework that captures the needs of customers, helps frame potential solutions through the analysis and definition of needs, promotes the generation of ideas and supports the selection and validation of candidate solutions through prototyping and testing. Design Thinking often features most prominently in the pre-project phase where considerable uncertainty and ambiguity prevails (e.g. discrepancies between what potential customers say and do). It is considered an essential tool in the creation of innovative portfolios and the selection of promising projects.
- **Lean** Concerned with the elimination of waste (known as *muda*) through the management of workflows and establishment of pull-based production systems enshrined in Gemba Kaizen and Kanban thinking. Lean places the customer firmly in the driving seat. Waste means any superfluous activity or useless outcomes (e.g. over-production of detailed requirements, development of unwanted features, software defects, delays arising from communication latency between interfaces). This school of thought has had a profound impact on Agile thinking where the use of its hallmark Kanban board (a laned chart in which items are placed depending on where they lie in the production process) is now commonplace. Lean advocates that workflows should be smoothed as measured by a cumulative flow diagram that captures the relative amount of work in each phase of the project. Thus requirements to be specified in detail are determined by constraints on development (i.e. only as much is developed as can be specified in the forthcoming timeframe). In turn these are determined by the capacity of testing (i.e. only as much is delivered as can be tested). Buffers are kept to a minimum throughout the process and work in progress is kept even.
- **Lean Startup** An approach to developing businesses under circumstances of extreme uncertainty that is based on Lean thinking, validated learning and innovation accounting. This requires a continuous approach to innovation where hypotheses are formulated and tested during solution development amid a build, measure and learn cycle. Experiments are created throughout to test minimum viable products aimed at establishing the acceptance and viability of the underlying business model, and their performance is assessed by tracking appropriate growth metrics.
- **DevOps** A set of practices that emphasizes the collaboration between solution developers, operations and other specialist IT staff (e.g. network engineers and database administrators) through an acknowledgement of their interdependence and a sharing of

common values (e.g. rapid delivery, use of feedback loops and measurement through metrics). This approach focuses on a pipeline connecting solution development to its operational usage, along which deliverables can be transported more and more frequently (e.g. several releases a day) and with greater control through the use of highly automated and dynamic infrastructure (e.g. integrated testing and quality control) collectively referred to as continuous delivery. Because it has a strong affinity with Agile and Lean thinking, DevOps seeks to break down the barriers that have traditionally existed between development, quality assurance and technology teams.

- **Beyond Budgeting** A set of accounting practices that collectively challenge the traditional notion of performance management based on budgetary controls, which are deemed inadequate to cope with dynamic and fast-changing environments. The core principles of Beyond Budgeting have much in common with the beliefs of the Agile community. Its process orientation advocates relative (rather than absolute) performance management on which team-based (rather than individual) reward systems are based, continuous and inclusive planning, controls based on relative indicators and trends, adaptive resource management and dynamic coordination.

1.7 Financial challenges

Since Agile activity is inherently adaptive, it bases decision-making on short-term certainties rather than long-term forecasting. Accordingly planning is often tiered in five distinct levels, as shown in Table 1.3.

Table 1.3 Five levels of Agile planning

Planning level	Scrum	DSDM
An overall vision of what is to be achieved by the solution	Product vision	Terms of reference, business case
Plan for the delivery of increments	Product roadmap	Delivery plan
High-level list of prioritized requirements and other tasks	Product backlog	Prioritized requirements list
Refined iteration plan comprising detailed requirements and tasks	Sprint backlog	Timebox plan
Daily planning of activities and handling of impediments	Daily Scrum	Daily stand-up

In challenging the traditional triangle of project management variables (i.e. time, cost and scope), Agile proposes that time be fixed and scope made variable under the assumption that cost is largely a function of time and materials. This creates a significant issue when faced with the budgetary cycle of planning where activities need to be planned over long periods of time (e.g. 12 months) using estimates from which costs are derived. Such

approaches owe their origins to plan-driven manufacturing and start to break down when resource-based, market-driven allocation is more appropriate. Instead budgetary control is reduced to a matter of measuring performance in terms of variation between planned and actual expenditures. It fails to take into account market movements and organizational learning. For example, long-term estimates that are the basis of budgetary planning are often founded on largely unvalidated assumptions concerning customers' needs (e.g. their inability to articulate what is required) or the technical feasibility of solutions (e.g. the learning required to be able to use a new technology). In the light of such uncertainty it would seem appropriate to engage customers early on to discover their true needs (e.g. through ethnographic studies, workplace shadowing or construction of needs hierarchies) before investigating solutions through prototyping and testing with users.

The manner in which budgets allocate funds for individual benefits, such as training, and performance-related pay (e.g. bonuses) also contribute to outcomes that sit uncomfortably with Agile thinking. For example, individual-based schemes have been shown to promote the withholding of information and inhibit collaboration in situations where individuals perceive a benefit in retaining exclusive specialist knowledge. Instead, by encouraging personal responsibility, knowledge sharing and self-organization within teams focused on the needs of customers, rewards can be more effectively used as a stimulus when applied in a group-based context. This encourages teams to work together towards collective goals and to act appropriately when faced with the unacceptable behaviours of a minority (e.g. social loafing, information withholding). Such environments are also conducive to the development of generalist skills that broaden perspectives and understanding, which in turn foster creativity and innovation.

Budgetary plan-driven approaches also suffer from the way in which financial resources are distributed incrementally based on cost centres (e.g. research and development, product development, procurement) rather than activities or market needs. This can create a culture of tribalism (e.g. defensive protection of annual budgets) and a premature framing of solution development approaches (e.g. a preference for an off-the-shelf solution if more funds are available in the procurement budget). Indeed, the manner in which the annual budget cycle is conducted seldom bears any resemblance to the prevailing market situation and often fails to exploit opportunities that arise. For example, Lean thinking, which aligns with the Agile approach in this respect, suggests that organizational units be created in response to market needs and be sized according to the target segment. Moreover, such units should actively test the feasibility and marketability of new product or service offerings through a collaborative and iterative process that engages potential future customers. This approach also enables an organization to disengage from an undertaking if the underlying assumptions are found to be invalid without fear of retribution or recrimination.

1.8 Concluding remarks

Agile accepts that change and uncertainty are inherent in product and service development and it encourages flexible organizational structures and adaptive practices. For this reason it is considerably more able to cope with uncertainty and unforeseen opportunities than traditional project management approaches. This needs to be acknowledged in project approaches which often use variance against a plan as the basis for assessment. The Agile approach structures the delivery and timing of value in order to maximize returns and to ensure continual validation. Because value is delivered in decreasing order, the returns on future deliverables are increasingly marginal (relative to the original baseline). Therefore, an incremental approach to appraisal is required, with decision points (concerning whether or not to continue with the project) being taken at an appropriate point after the delivery of each new increment. A key element of Agile financial appraisal, derived from Lean thinking, is that the learning that takes place during an Agile project helps to reduce future wasted effort and should therefore not be considered a write-off attributable to poor initial planning. Moreover, new insights gained along the way usually result in internal restructuring of the solution that reduces future costs of maintenance (i.e. technical debt) or the need to overhaul the entire solution at a later point (i.e. major refactoring). All this leaves the traditional hallmarks of project appraisal (i.e. the use of long-term projected cash flows and the delegation of maintenance costs to operations) looking rather out of place in a dynamic and uncertain world.