Alan Moran

Managing Agile

Strategy, Implementation, Organisation and People

Springer

Managing Agile

Alan Moran

Managing Agile

Strategy, Implementation, Organisation and People

Springer

Alan Moran
Zurich
Switzerland

ISBN 978-3-319-16261-4 ISBN 978-3-319-16262-1 (eBook)
DOI 10.1007/978-3-319-16262-1

Library of Congress Control Number: 2015934045

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

This book is dedicated to my three brightest shining stars Helen, Markus and Patrick!

Preface

This book considers agile from a management perspective by focusing on matters of strategy, implementation, organisation and people. It examines the turbulence of the marketplace and business environment in order to identify what role agile has to play in coping with such change and uncertainty. What emerges is a narrative concerning a new strategic orientation based on dynamic capabilities and the renegotiation of meaning within the organisation that lends itself to implementation using appropriate agile project and programme management techniques. This results in a transformative experience for the organisation which must challenge long held beliefs about its own identity and embrace the entrepreneurial spirit of its employees. Accordingly self-organisational paradigms directed outwards towards the needs of customers and teams that possess the necessary autonomy and empowerment become the new order of play. This book, based on observations, personal experience and extensive research endeavours to make clear the fabric of the agile organisation thereby assisting managers to become agile leaders in an uncertain world.

This book is intended for those with decision-making authority within their organisations (e.g., line managers, programme, project and risk managers, senior managers) about whom the assumption is made that the essentials of topics such as strategy, finance, quality, governance and risk management are not only understood but constitute a daily aspect of their working lives. It is also likely that this book will be of interest to those studying advanced management or business administration courses (e.g., M.B.A., M.Sc.), who wish to engage in the management affairs of agile organisations and thus need to adapt their skills and knowledge accordingly. Though operational aspects of agile (e.g., how to create a backlog of requirements or manage a facilitated workshop) are occasionally alluded to they are not the main concern of this book as such matters are covered at length elsewhere in the wider agile literature.

This book opens with a broad survey of agile contrasting and comparing some of the major methodologies selected on the basis of where they lie on a continuum of ceremony and formality ranging from the minimalist technique-driven and software engineering focused XP, through to the pragmatic product-project

paradigm that is Scrum and its scaled counterpart SAFe®, to the comparatively project-centric DSDM. Each enjoys the following of a vibrant community and all are well documented in detail elsewhere. Though reference is made to each of them, DSDM occupies a special place in this book owing to its comprehensive elaboration of programme and project management practices.

Discussion then moves on to the formulation of agile strategy framed in the language and concepts of the dynamic resource-based view and institutional theory schools of thought that best capture the attitude towards change and uncertainty that agile seeks to embrace. The relationship between agile, innovation and entrepreneurship is also elaborated on before considering matters of financial budgeting and appraisal and the contribution that agile makes to returns on investment and ultimately the bottom line.

A detailed discussion of agile programme and project management using the DSDM frameworks as their basis sets the stage for a series of topics including governance, quality, risk and configuration management. This analysis makes clear the link between strategy and its implementation transposing practices that may already be familiar to readers with a more traditional background in the agile context. Accordingly, there emerges not only a transition path towards more agile ways of working but also explanation of the specific nuances that distinguish traditional and agile approaches to programme and project management.

This book concludes with an analysis of the implications of agile for the wider organisation and a discussion of people factors. It is owing to the central organisational paradigm of self-organised teams that these two spheres are intimately linked. Indeed, self-organisation becomes the arena in which many people aspects find their expression (e.g., autonomy, flexibility) supported by wider organisational learning from which team performance is ultimately derived.

Whilst this book may be read sequentially many readers may find that they can skim material already familiar to them choosing instead to focus on specific details where appropriate. Indeed, the material has been divided into broadly self-contained parts so that readers may opt to focus their reading accordingly. In any event, the impatient reader may consider first reading the "Management Implications" section of any chapter only delving further into details should they encounter something that piques their interest. Since this book is based on extensive research, additional information can be found in the inline citations or in the hundreds of references found in the bibliography.

Acknowledgments

This book is based on practical experience and personal reflections validated by exchanges with a wide range of practitioners, leading figures and researchers within the agile and management communities. It builds upon the learning and wisdom of others who have experienced both success and failure in their application of agile. Thanks must be extended therefore to all who directly or indirectly contributed to this book through their discussions, feedback and thoughtful insights. Special thanks, however, is afforded to the reviewers Arie van Bennekum, Mirosław Dabrowski, Julia Godwin, Judy Graham, Claudia Lienert, Myriam Harmed Torres, Sridhar Nerur, Michael Nir, Simone Onofri, Konstantinos Panagiotidis, Barbara Roberts, Luis Sequeira and Richard Walker whose remarks and comments spoke volumes about their depth of experience.

Contents

Chapter 1
Introduction

Abstract Agile, originally derived in part from the manufacturing sector, has evolved into a set of principles and practices that have flourished within and found applications beyond the IT sector. Its adaptive, value-driven, collaborative and empowering essence drives innovation in an iterative and incremental manner that is founded upon organisational and experiential learning. Over time, many methodologies (e.g., XP, Scrum, DSDM, SAFe®) have become established reflecting different facets of Agile ranging from engineering, product development, project management and enterprise architectural perspectives all of which continue to be influenced by other developments (e.g., lean). Assessed in the sober light of day, however, Agile not only poses but also faces challenges within organisations that must question the role of existing structures and hierarchies, people factors together with their culture and processes that ultimately address the core nature and existence of the enterprise.

1.1 Agile Defined

Agile, as a concept, came to prominence in the late 1990s through the endeavours to address perceived difficulties with existing solution development processes that were rooted in, and owed their rigidity to, plan-driven practices. The seeds of Agile had been sown in the 1980s when the shortcomings of traditional methods were becoming more evident prompted by the rise of new technology and the increasing volatility of the business environment. Advocates of Agile promoted the notion that project uncertainties should be embraced and sought to balance planning and control with execution and feedback. Accordingly, agile projects exhibit features of open communication amongst heterogeneous stakeholders, emergent behaviour within self-organising teams and a culture of openness and learning [50]. Central to agile software development are the notions of iterative development and incremental delivery based on shared values enshrined in the agile manifesto (reproduced for ease of reference in the Appendix A) which expresses preferences towards individual interactions and customer collaboration, working solutions over comprehensive documentation and responsiveness to change. Thus the manifesto articulates the beliefs that interactions amongst project team members and their customers should

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_1

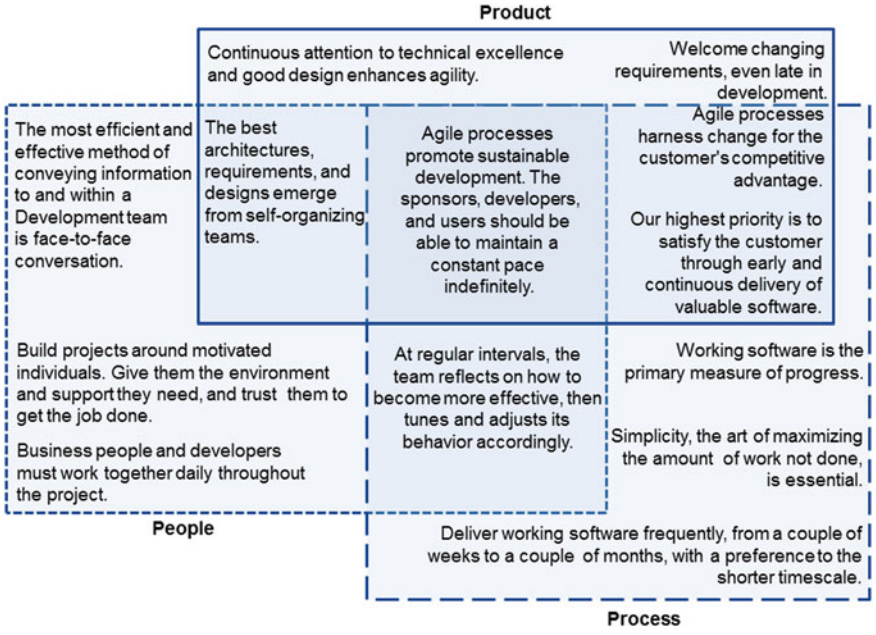

Fig. 1.1 Agile Principles in terms of Product, Process and People. Published with kind permission of © Alan Moran 2015

underpin efforts to create working solutions[1] in a flexible manner. Equally important are the twelve principles behind the manifesto that emphasize continuous delivery, an attitude of embracing change, frequent delivery of functional components, daily interaction with business stakeholders, empowerment through trust and support, direct communication, measurement of progress through functional software, sustainability, continuous excellence of design, simplicity through minimalism, self-organisation and team reflection. Understood as statements that guide action, the twelve agile principles are indeed found to all satisfy this definition of principle [238, 239]. Figure 1.1 depicts one possible analysis of the agile principles through the perspectives of product, process and people. This reveals that most principles relate strongly to one of these perspectives whilst others impinge significantly on at least two. The welcoming of change principle (sometimes referred to as the "embrace change" principle) contains subelements that can distinctly be assigned to both product and process domains whereas sustainability of development is arguably the principle that is common to all perspectives.

Defining Agile is, however, harder than it seems since much of what practitioners understand to be the essence of Agile arises from principles and practices experienced

[1] Although formulated in terms of software development, there is prevailing consensus amongst agile leaders that were the manifesto to be rewritten today then the word *software* would be replaced with *solution*.

as emergent characteristics. Thus agility can hardly be attributed to specific individual techniques or rituals since it is widely accepted that many of these existed prior to the agile movement though perhaps an argument can be made for the specific combinations that Agile promotes (e.g., test-driven development, continuous integration and refactoring definitions of which together with other agile techniques can be found in Appendix B). Nor can Agile be characterised purely by reference to the agile manifesto and its principles since some principles are likely to prove troublesome if used as the basis of a definition. For example, if emphasis is placed on face-to-face communication as suggested by one of the principles then teams that are not co-located could not be said to be strictly Agile. Ultimately, what emerges is agile as a discipline that copes adaptively with rapid change through feedback learning loops that iteratively create and incrementally deliver value. Accordingly, Agile can be understood as a loosely structured solution development paradigm that embodies the following core elements surrounded by a regular cadence of iterative development and incremental delivery driven by business needs (see Fig. 1.2):

- *Adaptive*. Accepting that change is inevitable and that the pursuit of reward entails risk, Agile advocates adaptive planning (i.e., high-level plans that are revised later into detailed plans once the necessary information becomes available) and effective feedback loops (e.g., reviews and retrospectives that guide and direct the solution development and the process behind it). This requires a decentralised and iterative approach to solution development that is responsive to the changing needs of business.
- *Value Driven*. By focusing on business needs, Agile promotes more direct assessments of progress (e.g., working solutions rather than status reports) and draws on the direct experience and input of those who need the solution. This requires tighter

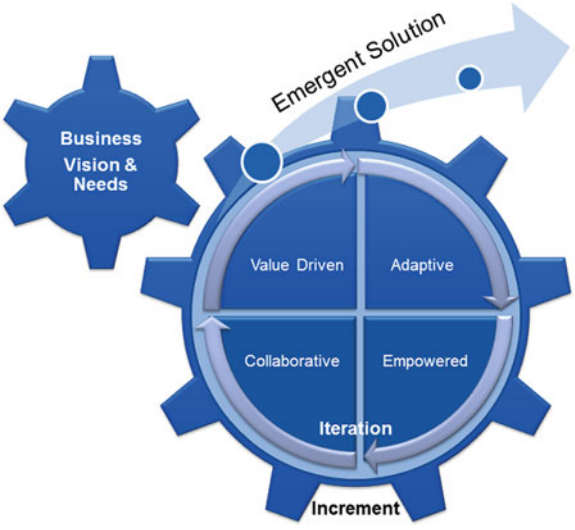

Fig. 1.2 Agile Solution Development Paradigm. Published with kind permission of © Alan Moran 2015

integration of stakeholders (esp. in relation to communication), leaner production practices and incremental delivery of working solutions.

- *Collaborative*. Agile eschews specialists working to specifications, preferring instead to employ multi-disciplinary and highly communicative teams that share their experiences and tacit knowledge in order to gain consensus regarding the solution (which may entail engagement of stakeholders outside of the team). Therein are found subtle nuances concerning how teams exchange information that place more value on direct rather than externalised interactions (see Chap. 10).
- *Empowered*. Establishing integrated teams requires humanistic values of trust, respect and courage supported by an environment of empowerment and self-organisation wherein the traditional role of management is replaced by one of servant-leadership.

More often than not, these core elements are found in practice to work in tandem. Agile is therefore as much a cultural stance on the process of solution development, as it is a set of practices and values. The discipline and focus needed to practice agility is frequently underestimated and has even been famously described as "hard and disruptive" by one of its leading proponents [235]. These ideas regarded ambiguity during the development process as a strength, provoking new perspectives and challenging established ideas, rather than a weakness that must be managed through precise planning. These agile principles find support in existing management theories already in widespread use. For example, the notion that motivation is increased if efforts lead to outcomes and outcomes lead to rewards that are valued is closely tied to the principles of building projects around motivated individuals and the promotion of a sustainable pace as described in Chap. 7 [264]. Elsewhere the concept of transaction costs economics (revisited in Chap. 6) poses the question if the notion of customer collaboration over contract negotiation places potential limits on the scalability of Agile since loose contracts admit opportunistic behaviours that create an imbalance between customer and provider [22].

Since value is a recurring notion in Agile it is worth considering for a moment what is meant by this term. In general what is expected of a solution is that it both performs a specific function (i.e., utility) and that it can be relied upon to do so a specific manner (i.e., warranty), the combination of which determines value in the eyes of the customer.[2] This combination of utility and warranty underpins the notions of fitness for purpose (i.e., the extent of actual utility of the solution) and fitness for use (i.e., the extent to which the intended purpose of a solution can actually be used) respectively. Specifically the extent to which a solution either improves an existing situation (e.g., faster transaction processing) or creates a new opportunity that did not already exist (e.g., support of new transaction processing paradigms) is a measure of fitness for purpose whereas its capacity to fulfill that function (e.g., reliability, security, availability[3]) describes its fitness for use. Fitness for purpose without fitness for use disappoints users who perceive the solution as inadequate

[2] This approach to considering value is common in service delivery frameworks e.g., ITSM.

[3] Availability as a measure of fitness for use is of particular importance in service provision (e.g., SaaS).

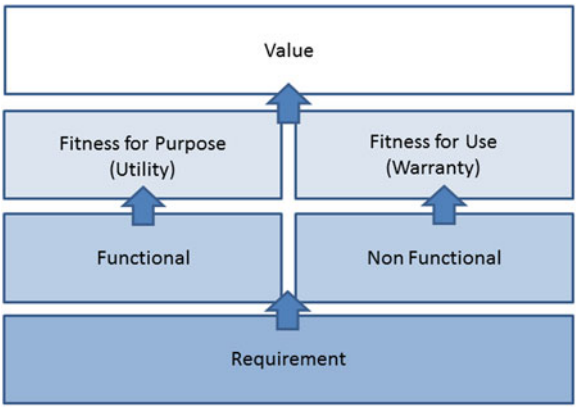

Fig. 1.3 The Utility/Warranty Relationship Between Requirements and Value. Published with kind permission of © Alan Moran 2015

in some respect (e.g., inconsistent, unreliable, unavailable, insecure) and fitness for use without fitness for purpose quickly makes a solution irrelevant. Ultimately it is down to the business analyst to capture those aspects defining value in terms of functional and non-functional requirements that underpin fitness for purpose and fitness for use respectively (see Fig. 1.3). Furthermore, since a solution is likely to entail both product and service components it is not only the solution quality but also the delivery capabilities of the organisation that will sustain value.

An alternative view of value may also be derived from information gathered from customers and used to determine the value drivers against which programmes and their projects can be assessed[4] [59]. In this sense, value represents the net effect of benefits derived against the costs incurred in acquiring them. The elegance of this focus on value, however, is not only the retention of its link to cost but also to risk and the alignment of planned activities with value delivery which lies at the heart of what Agile is all about.

1.2 Historical Development

The origins of Agile are rooted in part in the Japanese manufacturing and industrial sectors to which many agile concepts owe their heritage. These include the visual control concept found in the Toyota Production System that later anticipated agile information radiators, the Kanban charts used in agile task assignment and tracking, and the continual influence of lean thinking on Agile today. The synthesis of Eastern and Western thinking so persuasively laid out by the authors that introduced the term "scrum", reflects the spirit in which the agile manifesto itself was conceived and points to a genesis founded in organisational learning and team empowerment [256].

[4] This definition of value is borrowed from the Management of Value framework which finds favour in traditional programme and project environments such as MSP®, M_o_R® and PRINCE2®.

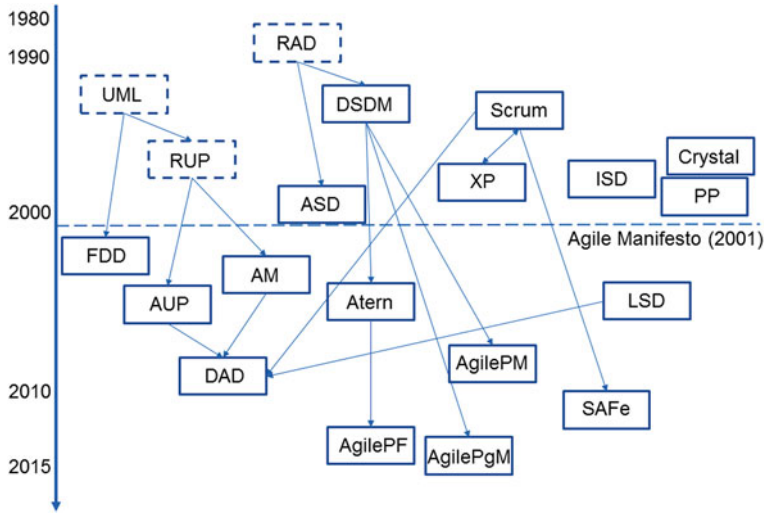

Fig. 1.4 Historical Development of Agile Methodologies. Adapted from [2]. Published with kind permission of © Alan Moran 2015

Figure 1.4 illustrates the historical development of the main agile methodologies since the late 1980s and early 1990s (the influence of traditional ways of thinking is indicated by dashed boxes).

In 1994 Dynamic Systems Development Method (DSDM), the first truly agile development method, emerged from RAD as an independent framework that has evolved over time to encompass a wider scope than one would traditionally associate with agile projects (e.g., inclusion of explicit governance, quality and risk) culminating in agile project management as described in the DSDM Agile Project Framework (AgilePF), the DSDM Agile Project Management (AgilePM®) and the DSDM Programme Management Frameworks (AgilePgM®). RAD employed iterative practices that drew upon the Spiral Model [29] and other ideas which had been practised by IBM by the 1990s [161]. By the takeover in 2003 of the Rational Software Corporation, however, Agile has already supplanted the term RAD and IBM saw its focus shift towards the Unified Process and its latter day agile variants. At about the same time, RAD also gave rise indirectly to Adaptive Software Development (ASD)[5] which actively sought to embrace change in speculate-collaborate-learn cyclical patterns of work [106]. In parallel Scrum, whose roots can be traced back to its industrial heritage in 1986 where it was conceived as a manufacturing development methodology that brought about "innovation continuously, incrementally and spirally", took shape from ideas stemming from organisational learning and went on to exert considerable influence within the agile world impacting almost all other methodologies [255]. At about the same time eXtreme Programming (XP), a software engineering

[5] RAD was initially developed as RADical software development before becoming ASD.

focused set of principles and practices that grew of the development of the Chrysler Comprehensive Compensation System payroll system, enjoyed a period of cross-fertilisation with Scrum setting the terms of reference for many agile practices today [25]. XP became hugely influential in the 1990s and its practices today constitute the essential tool set of every IT project that employs an agile approach. Around the time of the formulation of the agile manifesto several other methodologies were to be found including the design and build focused Feature Driven Development (FDD) as described in [200]. Internet-speed Development (ISD) [21, 58, 208] which practiced fast release and delivery, the practical toolkit that constituted Pragmatic Programming (PP) [122] and the flexible Crystal family of methodologies [50]. The efforts to make agile the Rational Unified Process (RUP) manifested themselves in the Agile Unified Process (AUP) which together with Agile Modelling (AM), focused on modelling practices and cultural principles, became the predecessors for Disciplined Agile Delivery (DAD) [8, 153, 237]. Alongside these developments other influential approaches arose including Lean Software Development (LSD) which applied lean principles to Agile [206] and the attempt to architecturally scale agile product development using the Scaled Agile Framework® (SAFe®) as detailed in [224].

This historical account reveals a rich texture of humanist (e.g., cognitive, social and interpersonal), organisational (e.g., managerial and cultural) and technological (e.g., practical and technical aspects) traditions for which there already exists a mature body of literature concerning both its culture and its practices [50, 52, 54, 66, 103]. Moreover, there are numerous comparative surveys of agile methodologies that highlight both the commonality rooted in the manifesto (and its principles) together with the uniqueness of focus and purpose with which each approach is endowed. Since the emphasis of this book is on the agile implementation of strategy in terms of programmes and their projects together with their impact on the wider organisation, DSDM will be used throughout as the primary reference for agile practices though on occasion details concerning other methodologies (e.g., Scrum, XP, SAFe®, DAD) will be delved into where appropriate.

As alluded to already, the DSDM Consortium was born in 1994 amidst efforts to harmonize software development practices around Rapid Application Development (RAD) suites that had, since the mid 1980s, been growing in popularity. By early 1995 the initial version of the DSDM framework, along with training and accreditation schemes, had been published with a second version to follow by the end of the year. By 1997 the emphasis was beginning to shift away from application development and towards business improvement process projects as a result of which a number of revisions to the methodology were made culminating in the release of its third version. In 2001 the DSDM Consortium was represented at and became a signatory to the agile manifesto. A fourth version was to follow in 2001 with two subsequent minor revisions appearing in 2002 and 2003. By 2007, when DSDM briefly flirted with the marketing name Atern®[6] (i.e., DSDM v5), it had for the first time become an open methodology that had assumed the mantle of a generic agile project management and solution delivery framework. Over this time several certifications came into being

[6] The portmanteau, Atern, is derived from the Artic tern whose characteristics reflect agile behaviour.

including the flagship agile project management qualification, AgilePM®, which today is accredited by the APMG.[7] In 2014 the DSDM Consortium launched the Agile Project Framework (i.e., DSDM v6), on which this book is based, which was followed by the Agile Programme Management methodology that scales to provide an agile means of managing programmes that deliver capabilities at the enterprise level.

With the introductory concepts and their historical background clarified, attention now turns to how agile practices are structured. A useful device in this context is an agile chart which captures the dynamic cyclicality of agile processes and enables communication of intent as well as solicitation of feedback. This also makes clear the distinction between iterations and increments that delineates development and delivery activities. Thereafter discussion of the practicalities of a generic agile process sets the scene for a detailed comparison of some of the major mainstream methodologies (i.e., XP, Scrum, DSDM and SAFe®).

1.3 Agile Charting

The iterative nature of agile projects is perhaps best expressed using agile charting, a technique that captures the dynamic of agile processes and which is used later in this chapter when comparing the mainstream methodologies [183]. Agile charts should be read in such a manner that each cycle of an outer circle implies multiple rotations of its inner circle. Thus in Fig. 1.5, which illustrates a generic agile process, each increment implies the traversal of one or more iterations which in turn involves the passage of one or more days. During each cycle the same tasks are repeated thereby conveying the cyclical nature of the process.

For reasons of simplicity a number of activities that ordinarily belong to pre- and post-project phases are omitted from Fig. 1.5. These include not only the formulation of the business case[8] but also the benefits review and other activities ancillary to solution development. Should the project context be sufficiently important then an additional outer cycle could be added though this is often omitted in practice. Note that the precise placement of activities on the cycles does not have any temporal implications (e.g., testing in Fig. 1.5 must not necessarily occur at the cycle halfway mark). Whilst it is likely that agile chart templates may exist at the enterprise level, they are intended to be tailored at the project level to reflect their specific vagaries. Indeed, recording what is actually happening in an agile project is often a good way to capture a baseline with which to frame process improvements. Some activities are linked at multiple levels thereby exhibiting recurrent characteristics as indicated by the slices in Fig. 1.5. For example, planning takes place at the release, iteration and

[7] Other certifications also exist albeit with less focus specifically on project management aspects such as the PMI-ACP® offered by the Project Management Institute.

[8] A business case is sometimes referred to as a "system metaphor" in XP or a "product vision" in Scrum.

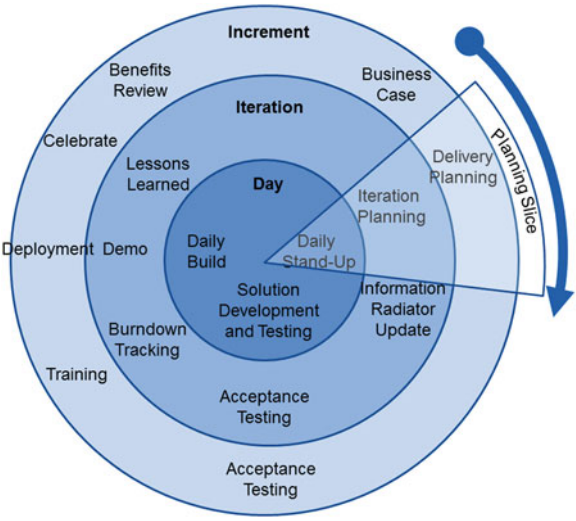

Fig. 1.5 Agile Chart for a Generic Agile Process. Published with kind permission of © Alan Moran 2015

daily (e.g., stand-up meetings) levels and is a good example of a recurrent activity. Deployment and testing related activities are similarly good slicing candidates.

Agile charts are a powerful management and communication instrument that can be used from the outset of a project to clarify when specific activities occur (e.g., stand-up meeting should take place at the start of each day or retrospective workshops are expected to occur at the end of each iteration). Agile charts also make evident any modifications or departures from the standard that one might wish to undertake in an agile process (e.g., annotating charts to indicate where risk management activities should occur within the process). Agile charting is a practice that should be understood both in methodological and project specific terms and is usually a combination of both. This means that whilst some activities reflect the practices of a chosen agile methodology (e.g., DSDM), other activities might reflect the specific environment of a project (e.g., the decision to deploy a nightly build asset into a centralised test environment). As a final remark agile charts can easily be commandeered for other purposes such as soliciting feedback during a retrospective, specifying quality gates for increments or indicating when tools are to be employed.

1.4 Iterations and Increments

From its earliest inception, Agile was understood as more of an evolution rather than a revolution. Its present day form is heavily influenced by the school of iterative development and incremental delivery that was by the 1980s already well established. Prior to this, Waterfall methodologies had epitomised the phased approach to

solution development[9] where each stage was expected to be completed before commencing with the next one and over time project management methodologies came to embody this approach.[10] The underlying belief was that change arose primarily from uncertainty during specification and that sufficient time spent detailing requirements up front would allay such concerns. It is interesting to note that the Waterfall method, which was adapted from the manufacturing industry, brought with it the assumption that changes late in the development process were prohibitively costly. This has been rigorously challenged by agilists who argue that postponement of design decisions until sufficient information is available is possible and endorse practices such as refactoring to cope with change.

In reality, solution development is more akin to organisational learning founded upon experience (incl. feelings and emotions), from which patterns can be identified (e.g., symbolism, narratives), leading to articulation (e.g., cognitive models and language) that results in application (e.g., product features). The experience-based learning that underpins this practical application is known as manifold learning [104]. This requires an entirely different approach built upon direct experience, collaboration, feedback and a culture of continual learning. One common misconception concerning Agile is that it can be understood as a phased approach "in the small" i.e., the repeated traversal of phases over a shorter time period (e.g., two to four weeks). This fails to grasp the essence of Agile since it misunderstands its inherently cyclical (rather than phased) nature and the manner in which previously distinct activities can become re-ordered or merged.[11] Between these two viewpoints of Waterfall and Agile, which were conceived approximately three decades apart, a plethora of variants can be found comprising both of plan-driven and humanistic influences. Indeed Agile finds itself today in a continual state of evolution with latter day influences emerging once again from industrial sector in the form of lean development practices [206]. Thus it was the iterative and incremental practices, atop of which Agile is built, that laid the foundations for feedback mechanisms that are used to cope with change and have become the hallmark of Agile.

For the purposes of this book, iterative development refers to the traversal (in whatever manner) of the entire solution development lifecycle (i.e., analysis, implementation and testing, deployment) with the aim of producing a self-contained, tested and partially functional product within a fixed timeframe. During successive iterations the product is further refined thereby enabling lessons learned from earlier iterations to be fed back into the process. Incremental delivery refers to the packaging and deployment of a product artefact (e.g., software solution, business process)

[9] One very common phased model is the Software (or Systems) Development Lifecycle (SDLC) which refers to a generic model of software development comprising of phases broadly divided into the soliciting and management of requirements, technical analysis and design, implementation of software solutions, validation and verification of software artefacts and concludes with the deployment and maintenance of the software solution.

[10] The PRINCE2® project management methodology promotes the notion of "technical stages" that contain similar types of work (e.g., analysis, implementation).

[11] A common example of the reordering of activities is Test-driven Development that advocates the creation of acceptance tests *before* (rather than *after*) the solution has been developed.

that can be meaningfully used by the customer, i.e., increments signal the intent or readiness, rather than the obligation, to release. Typically an increment will require several iterations to complete and several increments are necessary before the final product is delivered. Borrowing from a common agile metaphor [201]. Figure 1.6 illustrates the iterative manner in which an artefact evolves. Though the general idea is clear from the outset, details emerge gradually over time allowing participants to incorporate feedback back into the production process. At each stage there is a gaining of understanding and an adding of detail which can be clearly demonstrated and delivered to the customer.

Some methodologies, notably Scrum, merge these two concepts into one referring to the process of "iterative and incremental development". Thus consistent with the above view of iterative development, Jeff Sutherland, a leading advocate of Scrum, describes iteration as the act of traversal of the whole process during each pass of which the product gradually comes into focus. For him increments are concerned with the notion that "incremental development is iterating on the whole thing" and that each iteration should conclude with a "minimal usable feature set that is potentially shippable" indicating that this implies that code must satisfy the following definition of done for the increment,

> thoroughly tested, well-structured and well-written code that has been built into an executable and that the user operation of the functionality is documented, either in Help files or in user documentation [234].

though he later concedes that he "could have been clearer on what'potential shippable software' means" [253]. There is thus a suggestion that incremental activity embodies the characteristics of a deployable artefact though Scrum leans towards the use of this term in the substantive rather than adjectival form e.g., each iteration "delivers a fully functional increment" [253]. In essence the use of the term, increment, reflects a conceptual designation that implies the potential release of a deployable artefact. In reality there is little consensus in the agile community concerning the precise definition of an increment or indeed where the boundaries of agility lie. For example, owing to the structure of most organisations, it ought not be assumed that a product development team has full control over the means of deployment or of the dissemination of documentation (e.g., corporate design, accessibility). Moreover, it is ordinarily the case that some deployment activities (e.g., training of

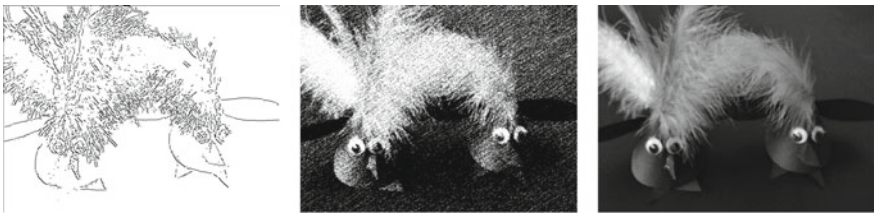

Fig. 1.6 Metaphor for the Iterative Evolution of a Solution. Published with kind permission of © Alan Moran 2013

users and service desk staff) are assumed by people outside of the project team. Whilst product focused methodologies (e.g., XP and Scrum) rarely consider what happens to the deliverable, other methodologies (e.g., DSDM, DAD, SAFe®) consider it within their remit to manage the entire delivery process from conception to delivery. Accordingly, this important conceptual distinction between the two related but different activities, iterative development and incremental delivery, is permitted accepting that the terms, iteration and increment, will have their own connotations within specific methodologies.

1.5 Agile in Practice

Agile teams tend to be small comprising of heterogeneous "generalising specialists" [9] capable of engaging in several distinct types of work (e.g., analysis, development, testing) [176]. Customer representatives are expected to be highly engaged, attend planning and demonstration events and be available on short notice should the solution team require their input. In this context the acronym CRACK (i.e., Collaborative, Representative, Authorized, Committed and Knowledgeable as mentioned in [32]), sets the tone for what is expected of such representatives. This is in stark contrast to non-agile approaches where business and development teams tend to be separated with relatively little contact beyond exchange of requirements and specifications. Agile methodologies employ a wide range of techniques and there is considerable overlap amongst the common methodologies both in their interpretation and application. Since the intention here is not to detail individual techniques and practices the reader is referred to the Appendix B for an overview. Some of these practices work well in concert (e.g., refactoring and continuous integration) whilst others might be considered complementary approaches to tackling a problem (e.g., modelling and prototyping). There does appear, however, to be a core set of practices that are used by most agile teams which comprises of daily stand-ups, iteration and release planning, unit testing, retrospectives, continuous integration, automated builds and burndown charting. These enjoy a common interpretation, though the precise wording may vary by methodology. Thus whilst it is true that all methodologies bear their own interpretation of the agile manifesto, it is equally fair to say that they borrow heavily from each other.

Generally speaking a team must balance the need for adaptation (e.g., innovation) against the potential pressure to standardise. This suggests that lightweight methodologies service high adaptation and low optimisation environments better, whereas heavier methodologies are to be found in low adaptation and high optimisation contexts [50]. It ought to be noted, however, that an enterprise typically requires both. Indeed it has been argued that the stability provided by a mature process oriented organisation creates the necessary conditions for an innovative agile environment to flourish [249]—a view that is perhaps at odds with the accepted wisdom within agile communities.

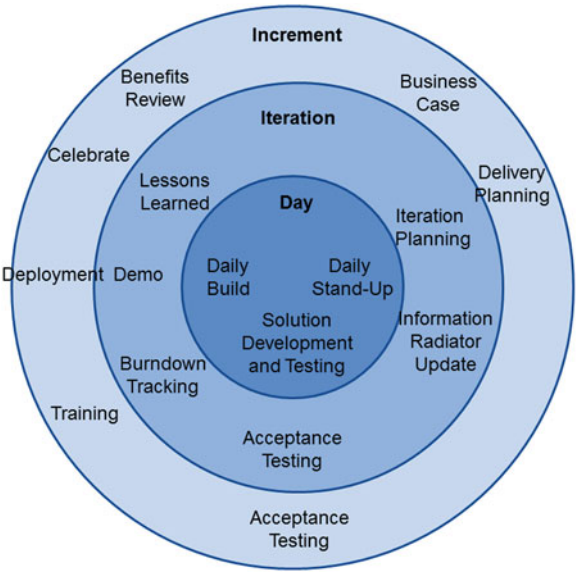

Fig. 1.7 Agile Chart for a Generic Agile Process. Published with kind permission of © Alan Moran 2015

Commencing with the inner cycle of Fig. 1.7, a typical agile day begins with a stand-up meeting of the project team members. During the day code may be developed, tested and integrated into a shared repository perhaps using continuous integration practices thereby ensuring a tight feedback loop (e.g., unit testing may be performed as part of the integration thus providing an early warning system for developers). At the end of each day a complete build and perhaps deployment may be performed to assess the stability and readiness of the code as well as demonstrating working software. Technical practices performed on a daily basis tend to be highly automated (e.g., continuous integration).

Continuing with the next cycle, each iteration begins with a planning session (which might be preceded by a "grooming" which is essentially a feature triage ahead of the main planning event) during which estimates and priorities are set [52]. Planning Poker, a simplified form of the Wideband Delphi method, is popularly used to gain consensus on estimates on the relative sizes of requirements (i.e., user stories). Throughout the iteration the communal information radiator may be updated with relevant information, user acceptance testing may be performed and progress tracked using some form of burndown charts reflecting the high degree of transparency often found in agile working environments (e.g., display of details on a communal information radiator). It may be that a Kanban board[12] is used to monitor the status of individual tasks as they progress through development and testing. Towards the end of the iteration the project team demonstrates the work completed during the iteration perhaps inviting wider stakeholders to participate. Concluding an iteration, the team reflects on its experiences and lessons learned (often referred to as a retrospective) and

[12] Kanban boards are referred to as Scrum-bans and Team Boards in Scrum and DSDM respectively.

considers what might be done to improve the process [66]. Iteration length typically varies between two to four weeks.

An increment requires affirmation of the business case and high-level delivery planning often resulting in a feature list (the term "backlog" was popularised by XP and Scrum though DSDM employs the term Prioritised Requirements List) that describes the requirements in the language of the customer at a level of detail commensurate with the information available (later subsets of this list becomes refined at the iteration level). Shared repository code may be isolated from the mainstream product code (a practice known as branching, itself a sensible risk management practice) though it is strongly recommended to instil a culture of frequent merging in order to avoid integration issues later. Acceptance testing of the evolving product usually occurs at all levels though its definitive character at the increment level is expected to complement the integration testing that occurs at the iteration and the unit testing in the daily cycle. Final deployment of a part (or complete) solution provides the welcome opportunity to celebrate delivery of business value!

1.6 Comparison of Methodologies

Today there are several well established agile methodologies in use, ranging from lightweight approaches that have a strong product development focus such as eXtreme Programming (XP) and Scrum to heavier methodologies such as the project management focused Dynamic Software Development Method or the architecture oriented Scaled Agile Framework® (SAFe®) as described in [25, 71, 203, 224, 233, 234]. Each methodology has its own culture, practices and language. For example, both XP and Scrum express their principles in humanist terms (e.g., respect, courage) whereas DSDM may appeal to those more comfortable in a mature environment with a stronger structural focus as evidenced in its formulation of values (e.g., focus on business need, deliver on time, collaborate, demonstrate control). Accordingly to gain an appreciation of their focus and sphere of application a brief survey of some of the major mainstream methodologies (i.e., XP, Scrum, DSDM and SAFe®) is warranted. These were selected on the grounds that they represent software engineering, product development, project management and portfolio architectural perspectives on Agile. The methodologies are compared on the basis of the five methodological dimensions of principles, roles, artefacts, practices and phases as illustrated in Fig. 1.8 and described in Table 1.1.

Needless to say each methodology elaborates on its practices to differing degrees of detail and thus a certain amount of interpretation is called for in order to make reasonable comparisons. For this reason, some caution is warranted when making direct comparisons based on the primary sources of these methodologies. For example Scrum describes the team (excluding the Scrum Master and Product Owner) as a role though strictly speaking the designation of organisational unit or function might

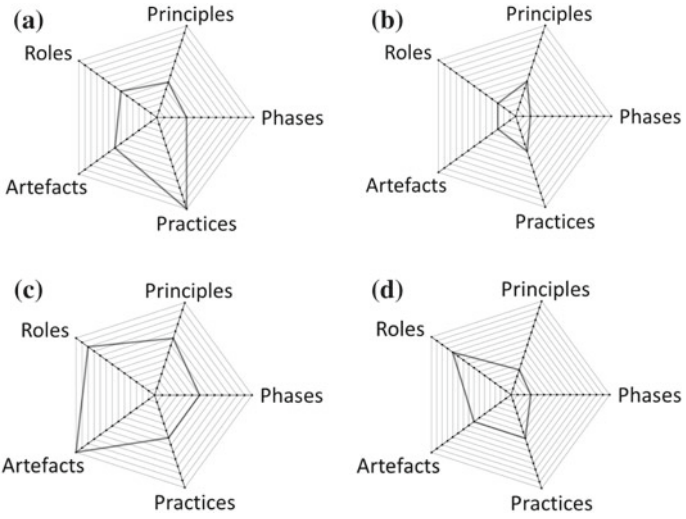

Fig. 1.8 Dimensional Comparison of Methodologies. Published with kind permission of © Alan Moran 2013. **a** XP Dimensions. **b** Scrum Dimensions. **c** DSDM Dimensions. **d** SAFe® Dimensions

Table 1.1 Definitions of Methodological Dimensions

Dimension	Description
Principles	The core values that characterise the methodology and imbue it with meaning in the eyes of its practitioners
Roles	Distinct roles cited by the methodology. Note that several roles may be assigned to an individual and thus no conclusions should therefore be drawn regarding team size
Artefacts	The intermediate products generated and consumed by the process (omitting final project deliverables). The necessity to create indirect artefacts is a reliable indicator of the weight of a methodology
Practices	The techniques explicitly cited by the methodology as being core to the effective and efficient operation of the process
Phases	The distinct phases of the model underpinning the methodology through which the process must traverse

be more appropriate.[13] Accordingly this analysis dips deeper using multiple sources to derive comparable figures.

A precursory glance at Fig. 1.8 indicates that there are substantial differences in terms of weight and style. In particular there is a visible division between the light-weight and technically focused approaches (e.g., XP and Scrum) and those that have

[13] In organisational theory a group that perform a specific task is referred to as a function. If formally embedded into an organisational hierarchy then the term organisational unit might be more appropriate. Either way a role would not be ordinarily assigned collectively in the manner suggested by Scrum.

aspirations to scale (i.e., DSDM and SAFe®). These reflect not only the audiences that each methodology wishes to address but also the historical roots whence they came. For example one major difference between XP and DSDM is methodological scope. DSDM embraces a wider set of concerns than mere product development and offers advice on a broad range of topics including governance, quality and risk. One topic that has received increasingly less attention since it was introduced in earlier DSDM versions is configuration management though this topic is covered extensively later in this book. This contrasts with the sharp XP focus on defining a tightly woven set of complementary and reinforcing practices centred around the production of code and the management of change. Irrespective of the specifics of any given agile methodology, however, there is always a working assumption of adaptation and tailoring to project circumstances.

1.6.1 Extreme Programming (XP)

The extreme programming (XP) model, see Fig. 1.9, draws heavily on existing practices taken to the extreme such as the writing of unit tests *prior* to implementing the solution in order to clarify what constitutes done and ensure immediate feedback on progress towards and validation of project tasks [23]. The result is a rich interwoven set of techniques that reinforced each other and today define the essence of an agile software development infrastructure. XP is a high discipline methodology that requires adherence to standards and a strong commitment to unit testing, refactoring and integration. The lack of focus on document output that characterises XP teams is an aggressive expression of the "working software over comprehensive documentation" statement in the agile manifesto. Indeed, XP eschews most forms

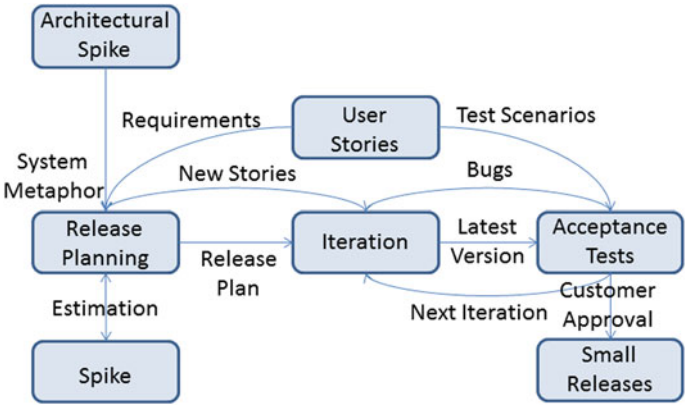

Fig. 1.9 Extreme Programming (XP) Model. Published with kind permission of © Alan Moran 2015

of administrative overhead (including reporting) and focuses entirely on software engineering practices. In general XP advocates generalists who can contribute to many facets of a project, working closely together and sharing their knowledge. The accusation levied against XP of being merely "old wine in new bottles" [108] seems unduly harsh since it fails to take into account the synthesis of practices and the cultural shift in software development that accompanied them. XP is therefore not without its critics though criticisms are often qualified on grounds that misunderstanding can lead to misapplication.

Table 1.2 lists an overview of the XP methodology in terms of the basic dimensional parameters introduced at the start of this section which have been reinterpreted in the interests of comparability. This confirms the technical focus of XP and whilst opinions concerning the precise nature of the other dimensions vary according to source, there is a broad consensus concerning what constitutes XP and what membership of an XP team might entail [25, 134, 271]. In particular XP is less model theoretic when compared with other methodologies and thus phases are inferred and constructed from descriptions of how practices are employed [270]. Owing to the fluid nature of XP, its interpretation in terms of an agile chart (see Fig. 1.10) should be understood as indicative rather than normative.

At first glance the distinction between increments, referred to as releases, and iterations (occasionally referred to as "Sprints") is somewhat optional in XP as evidenced by the following remark though the conceptual separation of release and iteration planning [269] is still retained:

> Some Agile projects don't even have iteration ends. They remain Agile by balancing the need for stable requirements with the need to change requirements while launching working software on a regular and frequent basis. ... iterations are just an easy way to demarcate when changes are accepted, when a new plan is created, and when working software is released to customers [272]

Table 1.2 XP Methodological Dimensions

Dimension	Description
Principles	Simplicity, Communication, Feedback, Respect, Courage
Roles	Customer, Analyst, Programmer, Tester, Coach, Manager
Artefacts	User Stories, Release Plan, Iteration Plan, Acceptance Tests, Documented Coding Standards, Vision (derived from System Metaphor), (Architectural) Spikes
Practices	Whole Team (incl. on site presence of the customer), Planning Game, Small Releases (i.e., incremental delivery), Customer Tests (incl. automated acceptance testing), Simple Design, Pair Programming, Test Driven Development, Design Improvement (i.e., refactoring), Continuous Integration, Collective Code Ownership, Coding Standards, System Metaphor (i.e., Vision), Sustainable Pace
Phase	Release Planning, Iteration, Acceptance Testing, Release

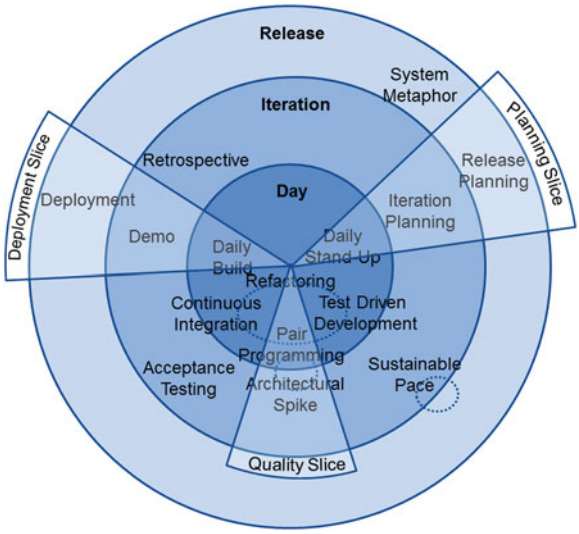

Fig. 1.10 XP Agile Chart. Published with kind permission of © Alan Moran 2015

Working inwards on Fig. 1.10 Releases are characterised as being where the vision (i.e., system metaphor) and team composition are determined (if these are not already agreed in the pre-project phase) and the point when high-level bundling of features together with the planning of their release (and hence the definition of sustainable pace) occurs. This leaves the iteration cycle concerned with the details of user stories (i.e., structured formulations of requirements) together with their prioritisation and estimation all of which takes place during the planning game (i.e., iteration planning session). It becomes necessary to augment this with modelling and prototyping which XP refers to as spikes (though at a higher level some consensus surrounding architectural aspects might have been decided at the release level as indicated by the equivocal bubble in Fig. 1.10). Acceptance testing that occurs at this level involves the customer directly and supplements the unit test testing that occurs on a daily basis. This cycle concludes with a demonstration of working software at the end of the iteration. Finally, daily activities revolve around the stand-ups at the start of each day (themselves a form of planning exercise) and the concrete practice of code ownership (e.g., pair programming) and validation of production (e.g., test driven development and continuous integration) which ensure that feedback is engrained in the process at a very deep level. That several XP practices are tightly integrated is suggested by the bubble connecting refactoring, test-driven development, pair programming and continuous integration. The concept of daily build and deployments is seldom explicitly cited in XP though it could be argued that it is present to a certain extent in continuous integration. Some elements of XP have been omitted in this description. For example it is possible that coding standards are best elaborated at the release level, in the pre-project phase or at the corporate level where the concept of collective ownership could also be supported. Moreover whilst retrospectives are an appropriate exercise at the end of each iteration, XP is not prescriptive about

when these should occur. In addition, simple design represents a guiding principle that ought to be elucidated at the iteration level and practised on the daily level.

XP exhibits natural slicing in respect of planning, quality and deployment (each illustrated separately in Fig. 1.10). For example when deciding in a specific project context at what level certain types of review exercises should occur (e.g., an innovative design idea) it is worth focusing on pair programming and (architectural) spiking to consider possible trade-offs. Equally the deployment process is well supported at all levels which clearly demonstrates working software whilst leaving open the option of where to embed specific deployment related activities (e.g., deployment of built artefacts into central corporate environments or practice of continuous delivery). Similar remarks apply to the planning and testing slices.

1.6.2 Scrum

Drawing on sporting metaphors the "relay-race" of the Waterfall methodology is contrasted with the "scrum" of holistic agility, which is based on empirical process control. Scrum practices are founded upon notions of inspection and adaption reminiscent of lean thinking. Scrum can best be described as a product development methodology with mild project management aspirations (e.g., lightweight tracking and reporting) as its focus lies in the management of software requirements and development. The scope of Scrum does not reach to other activities such as business change management, systems development or data migration and it defers to existing practices within an organisation to cover project initiation, risk management, release and deployment and change management processes. Indeed agile methodologies that have a wider scope have argued that Scrum can be successfully embedded within their frameworks [56, 124]. Scrum shares a common heritage with XP and both employ similar terminology and practices. Differences are, however, apparent in their structure and philosophy. For example, whilst XP focuses on software engineering practices, Scrum structures the development process (see Fig. 1.11) and is less prescriptive on matters of technique (though it does impose a variety of process constraints concerning team size, iteration duration etc.). Through the sharing of ideas and the embracing of heterogeneity of product teams, personal knowledge is encouraged to become organisational knowledge. It is interesting here to observe

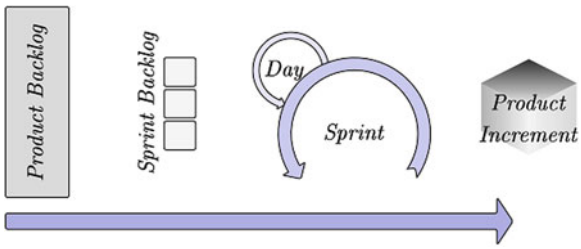

Fig. 1.11 The Scrum Process Model. Published with kind permission of © Alan Moran 2013

Table 1.3 Scrum Methodological Dimensions

Dimension	Description
Principles	Focus, Courage, Openness, Commitment, Respect
Roles	Product Owner, Scrum Master, Development Team
Artefacts	Product Backlog, Sprint Backlog, Iteration
Practices	Sprint, Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective
Phases	Sprint, Release

the merger of lean thinking, also taken from Japanese industry, resulting in the use of Kanban boards (co-opted in Scrum circles as an optional "Scrum-ban" though Scrum itself is not prescriptive in terms of techniques that must or even ought to be applied), a topic returned to later in this chapter. Today Scrum enjoys widespread popularity and its maturity is reflected in the size of its community and the automated tool support of its practices.

Scrum is considerably understated in terms of methodological dimensions and though this account remains faithful to the mainstream literature (see Table 1.3), it is noted that additional elements are often necessary. For example, Scrum designates as a developer[14] anyone who is neither a Product Owner nor a Scrum Master "regardless of the work being performed" [233]. Other methodologies differentiate roles more finely and it is to be assumed that most Scrum teams do so in practice. Table 1.3 shows the apparent lightness of Scrum based on summative information wherein values rather than principles and events rather than practices are referred to. As will become evident, Scrum focuses very much on providing a framework for development practices along with a means for reporting and monitoring progress. This added structure comes at the cost of the flexibility afforded by XP but replaces it with the freedom to adopt whatever techniques the team deems appropriate.

Figure 1.12 is an agile chart interpretation of Scrum based on multiple authoritative sources [203, 233, 234]. The distinction between increment (referred to as a release) and iteration (referred to as a Sprint) is relatively clear in Scrum based on the planning distinction between Product and Sprint backlogs which can be used to define the boundaries between the two. Thus it is at the release level (if not at the pre-project level) that the product vision is established and the coarse grained planning, in the form of epics recorded in the product backlog, occurs.

Sprints begin with a planning session which may employ collaborative techniques (e.g., Planning Poker, a simplified form of the Wideband-Delphi technique) and result in the creation of estimated and prioritised user stories grouped into epic categories that were created at the release planning level. In the event that user stories are assigned story points [52] the project velocity is expressed in terms of points per Sprint, which given the fixed timebox nature of a Sprint, indirectly translates into a point-to-day exchange rate determined by the slope of the burndown chart (which graphically depicts the remaining points as a function of time). Completion criteria

[14] Sometimes the function designation "The Team" is used.

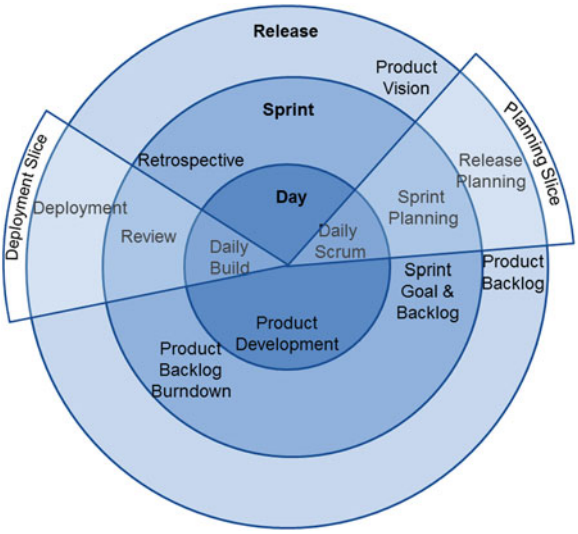

Fig. 1.12 Scrum Agile Chart. Published with kind permission of © Alan Moran 2015

are expressed at the level of Sprint backlog items and for the Sprint itself a definition of done is formulated. Sprint productive work concludes with a review to "foster collaboration and elicit feedback" [233] which, despite claims to the contrary, has something of the character of a status gathering meeting. It is during the review that the work completed in the Sprint is demonstrated. Note that it is permitted to challenge the value proposition of the product as well as the contents of the Product Backlog in this meeting. The review is followed by a retrospective which endeavours to identify means to improve the process, a practice which together with Scrums and Scrum-bans[15] illustrates the transparency that Scrum promotes. Feedback should be planned and integrated in time for the next Sprint. Each day starts with a structured stand-up meeting, referred to as a Scrum, wherein progress since the last meeting, plans for the day and impediments are discussed. Team members are understood as the primary active participants though it is not uncommon for other passive participants to be in attendance. Scrum has only one natural slice in the form of planning though others are likely to become apparent within specific projects once their practices are decided upon. The fact that there are fewer slices compared to XP simply reflects the framework character of Scrum.

1.6.3 Dynamic Systems Development Method (DSDM)

Rather unusually for an agile methodology, DSDM embraces a very wide spectrum of activities throughout the project lifecycle explicitly citing topics such as governance,

[15] Scrum-ban is an adaptation of Scrum which includes a tracking practice borrowed from Kanban.

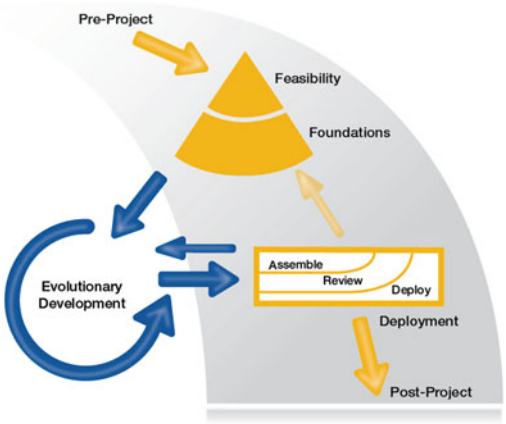

Fig. 1.13 The DSDM Process Model © DSDM Consortium 2014. Reprinted with permission

quality and risk. Accordingly it is rather more likely to appeal to those in a mature corporate environment who are already accustomed to the weight of ceremony[16] implied by the methodology as well as those seeking to link agility to governance and agile portfolio and programme management [70, 72]. This weight of DSDM may in part be due to the requirement that it expresses compatibility with lesser agile industry frameworks, a decision that was consciously taken at its inception in 1994. Whilst the number of artefacts may appear overwhelming to most agilists, DSDM is at pains to stress that these do not constitute full documents in the traditional sense stating that:

> Substituting traditional 'big design up front' with DSDM's 'enough design up front' promotes Agility in developing the required solution whilst avoiding the risk 'no design up front' that makes many larger and more strongly governed organisations so nervous [73].

The six phase model of DSDM is endowed with a rich set of roles and artefacts making it suitable for corporate project environments with established portfolio management and governance practices in place that may be willing to embrace agility but who seek reassurance that control can still be retained. Indeed the manner in which projects are structured ought to be reasonably familiar to those working with traditional project management methodologies (e.g., PMI, PRINCE2®). The DSDM model, depicted in Fig. 1.13, stipulates a feasibility and foundations phase followed by evolutionary development (earlier versions decomposed this phase into Exploration and Engineering phases) activities that culminate in one or more deployments.

At a glance it is clear from Table 1.4, that DSDM is very detailed in the guidance it offers. Note that the additional roles of Workshop Facilitator, DSDM Coach and other specialists are considered supporting roles in DSDM and are therefore not included in this list. Moreover for reasons of comparability with other methodologies some artefacts are not cited in this list (e.g., Evolving Solution, Deployed Solution).

[16] Though DSDM is rich in advice and artefacts it is important to note that not everything is considered mandatory.

Table 1.4 DSDM Methodological Dimensions

Dimension	Description
Principles	Focus on the Business Need, Deliver on Time, Collaborate, Never Compromise Quality, Build Incrementally from Firm Foundations, Develop Iteratively, Communicate Continuously and Clearly, Demonstrate Control
Roles	Business Sponsor, Business Visionary, Project Manager, Technical Co-ordinator, Team Lead, Business Ambassador, Business Analyst, Business Advisor, Technical Advisor, Solution Developer, Solution Tester
Artefacts	Terms of Reference, Business Case, Prioritised Requirements List, Solution Architecture Definition, Development Approach Definition, Delivery Plan, Management Approach Definition, Feasibility Assessment, Foundation Summary, Timebox Plan, Timebox Review Record, Project Review Report, Benefits Assessment
Practices	Facilitated Workshops, Timeboxing, Iterative Development, Moscow Prioritisation, Modelling (incl. Prototyping)
Phases	Pre-Project, Feasibility, Foundations, Evolutionary Development, Deployment, Post-Project

DSDM describes the development process in terms of cyclical execution of activities that lead iteratively and incrementally to the final solution and offers an option to structure the iteration cycle, which it refers to as a Timebox, to act as a control mechanism. Thus much of the emphasis of the model is on demonstration of control and providing a comprehensive structure around project activities.

From Fig. 1.14 it can be seen that an increment is characterised by the delivery of a deployment unit (though whether or not an increment constitutes a release is left

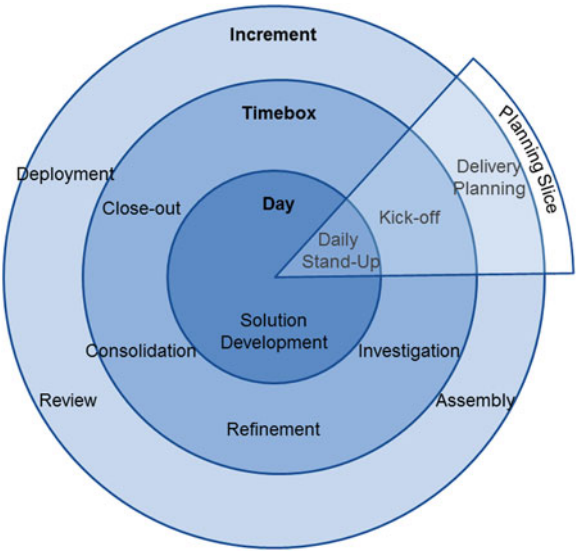

Fig. 1.14 DSDM Agile Chart (based on Structured Timeboxing). Published with kind permission of © Alan Moran 2015

open) and involves planning activities leading to the deployment phase comprising of assembly, review and the actual deployment itself. Increments are created by one or more Timeboxes which comprise of Investigation, Refinement and Consolidation steps, which DSDM refers to as iterations, sandwiched between short kick-off and close-out sessions. DSDM defines the Iterative Development process as an initial conversation, followed by repeated informal cycles of Thought, Action, Conversation. By creating Timebox Review Records, which can be either informal or formal, DSDM allows for the auditing activity which may be applicable or even mandatory in some project contexts (e.g. compliance within a regulatory environment) and this can be said to be an extension of agility to domains more commonly associated with traditional project management methodologies. Beyond occasionally mentioning a stand-up meeting and development efforts, DSDM is rather silent about structure at the daily cycle. This might be attributable to the fact that DSDM sees itself as providing the necessary framework for oversight and project management without stipulating the precise nature of production activities. It is in this respect that one of the sharpest contrasts between XP and DSDM can be seen both of which occupy different branches of the product-project dichotomy.

1.6.4 Scaled Agile Framework® (SAFe®)

The Scaled Agile Framework® (SAFe®), see Fig. 1.15, describes itself as an "interactive knowledge base for implementing agile practices at enterprise scale" [224] and cites a number of scalable practices that must be mastered in order to achieve enterprise agility. These include the definition and organisation of committed teams tasked with building and testing components that are accountable for delivering results with dual levels of planning and tracking (akin to the separation of iterative development and incremental deployment), a mastery of the iteration as the "heartbeat of agility", smaller and more frequent releases (in tune with lean thinking), concurrent testing and integration and a culture of reflection and adaptation [151].

There is an emphasis on the atomic nature of requirements and design that makes a strong argument for their continual evolution during the development process on grounds that neither has any meaning without the other rather than separating them into distinct phases. Institutionalisation of continuous integration is supported by efficient project structures that draw heavily on XP practices. This reflects a wider appreciation within the agile community of continuous feedback that today has come to be understood as involving much more than just testing (e.g., automated code quality analysis and security reviews). Proposing a hybrid matrix organisation the case is made for teams that draw the right people from disparate functions and departments (e.g., product definition, software development, testing and quality) to achieve an agile organisation of staff. This is reflected in the "servant leadership" role of team leads who act as facilitators empowering those within their teams. Organisational structures range from Scrum of Scrums who track progress on a daily basis to heterogeneous steering committees comprising not only of team leads but also architects,

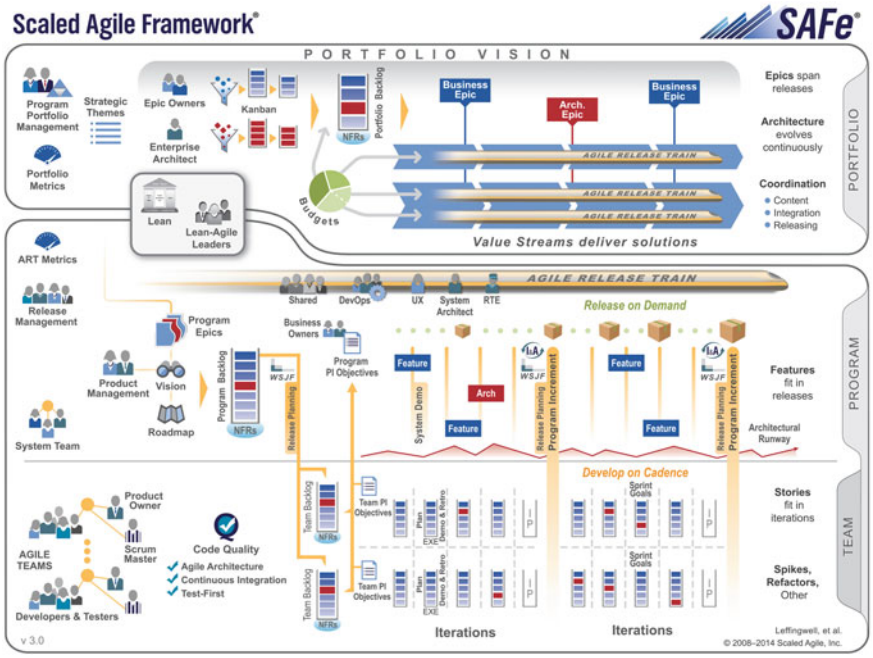

Fig. 1.15 The SAFe® Big Picture. Reproduced with permission from © 2011–2014 Scaled Agile, Inc

managers and other stakeholders as appropriate. SAFe® challenges the notion that agility is confined to the realm of small co-located teams based purely on emergent architecture that are light on requirements analysis. Rather it suggests that at scale, all development is distributed development and that agility must therefore rise to this challenge by scaling it practices and incorporating specific techniques (e.g., forward-looking architecture referred to as an "architecture runway"). It acknowledges the impediments that some organisations place in the way of agility such as resistance to change born of a desire to protect established assets, introduction of controls that subtly introduce Waterfall style gateways and reward systems that favour the individual over the collective. The practices of SAFe® find integration in the creation of an agile enterprise founded on intentional architecture arising from largely independent components with clearly defined interfaces, a lean approach to requirements based on product visions and just-in-time elaborations, employment of functional variability to coordinate product releases, appropriate tooling to support highly distributed teams, organisational and performance measure adaptations (e.g., agile balanced score cards), some of which have already been alluded to above.

Table 1.5 depicts the methodological dimensions of SAFe®. Similar to the treatment of Scrum and XP, principles in SAFe® have been identified with underlying core values. Roles can be categorised at multiple levels though those that more accurately

Table 1.5 SAFe® Methodological Dimensions

Dimension	Description
Principles	Alignment, Code Quality, Transparency, Programme Execution
Roles	Business Owner, Developer, Enterprise Architect, Epic Owner, Product Owner, Product Manager, Release Train Engineer, Scrum Master, System Architect, Tester, UX Designer
Artefacts	Epics, Portfolio Backlog, Portfolio Vision, Program Backlog, Release Plan, Roadmap, Strategic Themes
Practices	Define/Build/Test Component Teams, Dual Level Planning and Tracking, Iteration Mastery, Smaller Frequent Releases, Concurrent Testing, Continuous Integration, Regular Reflection and Adaptation
Phases	Iteration, Release, Value Stream

describe functions[17] have been omitted from Table 1.5 which leaves those that enable a reasonably accurate comparison with other scaled methodologies (e.g., DSDM). SAFe® promotes a wide range of practices and artefacts though those that are cited here reflect the essence of what is claimed to be necessary in order for Agile to scale. An agile chart for SAFe® strongly resembles that of Scrum (see Fig. 1.12) at the Team level (as identified in Fig. 1.15) beyond which phased organisational structures are mostly to be found.

1.7 Systems Thinking

Systems thinking refers to a discipline that suggests that systems are better understood in terms of subsystems and the dynamics of the relations that exist between them rather than through repeated reductionist analysis [229, 236, 240, 266, 268]. It has grown into a popular management perspective encompassing a wide range of topics (e.g., quality management, organisational learning). Figure 1.16a depicts a simplified systems perspective of the solution development process that identifies subsystems aligned with the role categories of DSDM (i.e., management, solution development and business as described in Chap. 4). In this case broad boundaries of functional discipline (e.g., technical or business skills) lie behind these subsystem definitions suggesting that the level of specialism is proportional to the number of subsystems that require coordination. Note that the act of defining subsystems is largely a matter of judgement that relies on the intent of the system representation. Indeed, this system cannot arguably have fewer subsystems since management seldom revolves entirely around a single product, and business units do not exist simply to have solutions developed for them. These subsystems together with the entire system in which they reside all belong to a wider environment which influences and interacts with the solution development process. Between any pair of entities

[17] In organisational theory a *function* denotes a team together with its resources that directed are towards specific processes and activities (e.g., release management).

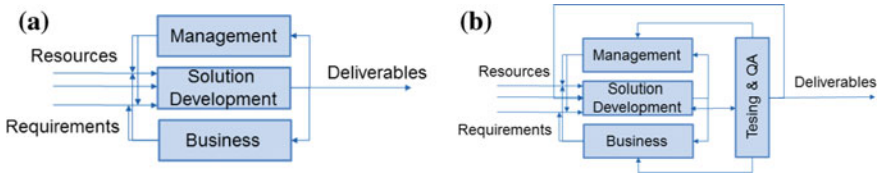

Fig. 1.16 Contrasting Topologies in terms of Systems Thinking. Adapted from [274]. Published with kind permission of © Alan Moran 2015. **a** Integrated Testing and QA. **b** Distinct Testing and QA

(e.g., subsystems or the wider environment) there are flows of information and artefacts reflecting the dynamic of the process some of which may be subject to delays that may impede the effectiveness or impair the responsiveness of the overall system (e.g., untimely information that elicits an inappropriate management response). Thus business provides inputs into solution development in the form of requirements and resources (e.g., access to Business Advisors) either of which may also arise from the wider environment (e.g., regulatory requirements or access to a DSDM Coach). In terms of outputs the solution team may deliver a variety of artefacts (e.g., software, processes) that feedbacks back into adaptive planning, value for business and impact the wider environment.

Systems theory would suggest that traditional environments comprising of highly specialised functions (e.g., separate QA and testing teams) require more subsystems that increase the delay of flows and become harder to manage in spite of the relative efficiency of the individual subsystems (see Fig. 1.16b). This added complexity requires that someone assume responsibility for coordination and communication and indeed this tends to be the primary justification for testing and quality assurance management roles. Agile environments, on the other hand, tend towards a reduction of separate subsystems that in turn reduces the latency of flow albeit at the relative cost of efficiency of specialists over generalists. This is alleviated to some extent by the enriching interaction between individuals and the malleable nature of the processes with which they work.[18]

Systems thinking identifies two strategies for the management of the flow of information around a subsystem: information stockpiling and closer integration [91, 236]. The former is endorsed by traditional approaches to solution development (e.g., full up-front specifications) which stands in contrast to the preferences of agilists who rely on comparatively lean inputs delivered in a timely manner (e.g., requirements as user stories prepared for the forthcoming iteration) to improve flow. Relieved of information stockpiles (e.g., detailed documentation and requirements catalogues), agile teams are also unencumbered with the efforts required to maintain them. It is therefore befitting the socio-technical and generalist nature of agile environments that separate functions are usually dispensed with in favour of integrated topologies.

[18] To what extent agile processes are truly malleable is a matter for debate. For example, methodologists of some of the mainstream product development and engineering have at various points in time insisted that nothing in the method ought to be compromised for the sake of the organisation.

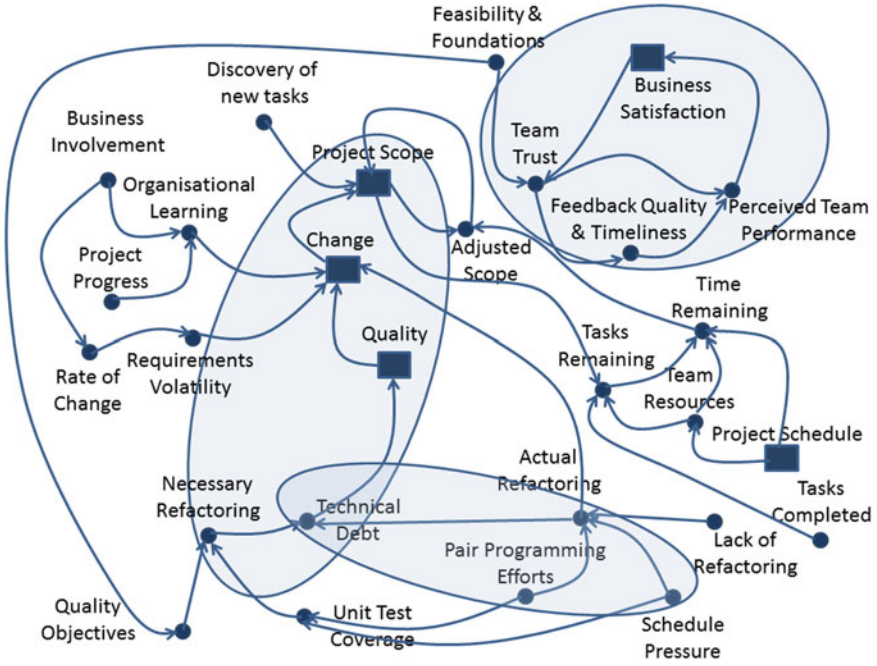

Fig. 1.17 Systems Dynamics Perspective on Agile. Published with kind permission of © Alan Moran 2015

Figure 1.17 illustrates the agile process in terms of key stocks such as business satisfaction, change, quality, schedule and scope (which are indicated as square boxes) and the flows that exist between entities involved in the process (adapted and simplified from though it does no justice to the detailed explanation provided therein) [39]. It reveals several important dynamical elements typically found in agile projects. These refer to the highlighted sections on the systems chart and include:

- The impact of business involvement on learning and the building of trust that feeds into both perceived performance and the quality and timeliness of feedback which in turn impacts business satisfaction creating a virtuous cycle between trust, performance and satisfaction.
- Understood in terms of adaptive, corrective and perfective change, agile processes suggest that refactoring is an appropriate mechanism for tackling both technical debt and overall solution quality and that such changes contribute to project scope in much the same manner as project schedule or the discovery of new requirements may. These three forms of change are defined as follows:
  - *Adaptive Change*. This refers to new requirements that are to be acted upon in the future unless they involve major alterations to scope, schedule or cost. The manner in which change affects project scope influences the rate of affective change.

  - *Corrective Change*. This change concerns alterations that require reworking of a solution and are usually based on review and feedback concerning an increment. The impact of technical debt on quality affects the rate of corrective change.
  - *Perfective Change*. This encompasses improvements and fine tuning of a solution. In agile environments is usually the focus of refactoring efforts. Indeed, the influence of the level of actual refactoring on the rate of change affects the rate of perfective change.

  Moreover, it may be assumed that the willingness to adjust scope is linked to the trust that already exists within the team since continual dialogue and negotiation are a central feature of agile processes.
- The link between test coverage, pair programming and refactoring towards the reduction of technical debt and the negative impact of schedule pressure (or the omission of refactoring) shows an important dynamic between these factors that ultimately impinges on quality. For example, studies have shown that individuals engaged in pair programming are more likely to improve test coverage and that under pressure individual developers will tend to dispense with writing unit tests. Similarly (lack of) refactoring efforts follow a similar pattern resulting in impaired code that may trigger the need for a major refactoring. This points to refactoring as a key contributor to the maintenance of quality in the long term. Indeed it may be concluded that it is the combination of these techniques (i.e., pair programming, test-driven development and refactoring) that contributes positively to productivity, quality and feature delivery.

As well as having been studied in systems thinking terms, Agile has also been treated as a complex adaptive system whose characteristics can be described as defying conventional modelling, exhibiting high degrees of non-linearity,[19] illustrating feedback loops with potential recurrency,[20] instability in terms of equilibrium that is driven entirely by local interactions which are largely ignorant of the system as a whole [107, 119]. The significance of complex adaptive systems is that they provide a theoretical basis for understanding how emergent effects can arise from change. This is far from obvious since intuition might dictate that for order to prevail there must be an overarching structure that brings it about.[21] However, the reality is that emergent order can arise from the locally simple interactions between subsystems. By analogy, in the absence of a pre-defined specification it is possible for members of a solution development team using simple local rules to determine a path towards a desired goal driven by emergent dynamic behaviours.

[19] Non-linearity refers to a disproportionate behaviour between input and output that deviates significantly from linear relationships.

[20] Recurrency, after a number of transitions, refers to the return to a past state and reflects a structural feature of some complex adaptive systems.

[21] Emergent effects are beautifully visualised by Mandelbrot sets which arise from simple but non-linear mathematically transformations of sets. Indeed this is the essence of chaos theory (e.g., the "Butterfly effect") which was very much in vogue when the agile manifesto was being formulated.

1.8 Influence of Lean

The terms lean and agile first found widespread use in the manufacturing industry and in particular in supply chain management where they concerned the matching of market cost and availability criteria respectively. In this context lean aims to manage workflows in order to level supply chain scheduling and to eliminate waste from the value stream whereas Agile focuses on using knowledge within the context of a virtual organisation to exploit opportunities in a volatile marketplace [192]. In general lean supply chains were exemplified by long product lifecycles, low profit margins and algorithmic forecasting whereas agile chains exhibit shorter product lifecycles, higher profit margins and consultative forecasting [164]. Owing to the punitive nature of supply outages and obsolescence of products in some markets, the need to dynamically align capacity to demand in a highly responsive manner arises. In particular, in the IT sector this need was serviced through fundamental alterations to the solution development process and in tighter integration of business stakeholders.

Lean describes a set of practices that aim to eliminate waste through the establishment of pull based production systems [281, 282]. Workflow management is at the heart of lean, which in essence is an extension of Just-In-Time thinking wherein orders initiate the flow of materials thereby matching supply and demand with low levels of inventory. In knowledge based industries this is interpreted as reducing the scope of development (i.e., focus on business need by avoiding the production of unused features), reducing information stockpiling (e.g., decreasing levels of granularity of requirements and specifications according to proximity to the solution development process) and reducing the number and complexity of interfaces (i.e., integrated and heterogeneous teams rather than use of separate specialist functions e.g., testing and quality). Adopting a user-based approach, lean emphasizes value from the perspective of the consumer of the value chain and seeks to eliminate actions or artefacts that do not contribute directly to value[22] [127]. Lean systems stress working with suppliers, application of quality control (e.g., Kaizen), reflection on process improvement (e.g., Five Whys technique[23]) and employee involvement. In general lean has been found to function best in environments where demand is both stable and predictable and variety is low [47]. Thus in IT no more is developed than can be tested and no more is specified than can be developed thereby levelling the process and creating uniform flow with low information stockpiles.

In contrast Agile concerns those matters that enable a team to cope with change. For example, self-organisation paradigms in which team members share information

[22] Kaizen processes, originally developed as part of the Toyota Production System, help to identify and tackle the seven sources of waste (i.e., transportation, inventory, motion, waiting, over-processing, over-production and defects).

[23] The Five Whys technique repeatedly poses why questions as often as is necessary (the number five is merely indicative) to get to the root cause of a quality problem. In fact the term *hansei* (i.e., reflection) is used in this context to indicate the retrospective nature of the exercise.

and exhibit collective ownership of issues in a spirit of positive collaboration and communication ensure that needs are promptly identified and addressed. Practices that promote feedback loops (e.g., daily stand-ups, continuous integration, retrospectives) reinforce this position and require adequate levels of flexibility and internal autonomy within teams. In addition, organisational learning (which may extend beyond the boundaries of the enterprise to encompass the building of partnerships) together with modular architectures and automated tool support and infrastructure are key enablers of Agile both of which assume strategic significance (as described in Chap. 2 in the discussion of anticipatory and reactive dynamics capabilities). This capacity to respond in a timely manner (often offering unique solutions) is an important indicator of the competitive advantage that Agile delivers and requires flexibility of process, organisation and culture.

Although within the manufacturing industry lean was generally perceived as the precursor to Agile, over time there has emerged a hybrid form, referred to as leagility. The precise mechanism by which this occurs varies but examples include Pareto partitioning of product ranges by demand (i.e., using lean strategies for those products that represent the stable majority of demand and employ Agile for the rest) or creation of a decoupling-point up to which generic modular components are produced in a lean manner and beyond which assembly and configuration follow agile practices. Within the IT and other knowledge-based sectors, however, the boundaries have been more blurred with direct evidence of the influence of lean on agile being found in some of its principles (e.g., maximisation of the amount of work not done) and practices (e.g., use of a Kanban board for workflow management or the pull characteristics of forced ranked backlogs). Indeed, common lean strategies reinterpreted in the context of IT solution development include the elimination of waste (e.g., manner in which requirements are managed or removal of obsolete code), pull-based systems (e.g., Kanban boards) and set-based decision-making (i.e., keeping open distinct design and integration options until such point in time as sufficient information exists to make a decision). Whilst perhaps less widely used than the mainstream agile techniques, lean has also resulted in the adoption of new techniques into the solution development process such as value stream mapping that differentiates between value and non-value adding activities in order to improve overall flow and value creation [206]. Thus it is fair to say that Agile as understood in solution development terms has evolved primarily as a change management strategy but has equipped itself along the way with practices inspired by a lean mindset.

1.9 Management Implications

Agile can hardly be considered new and today occupies a prominent position in the IT sector with leading industry analysts predicting that increased productivity is linked to team flexibility and the nature of coping mechanisms that deal with changing requirements [189]. Perspectives on Agile are, however, coloured by the methodology

that an organisation chooses to implement and vary from product and scaled architectural approaches to full project and programme management frameworks. However, the suspicion that Agile is a purely operational affair with its own technocratic practices must give way to its identification as a vehicle for strategic and organisational change. Indeed, in spite of its manufacturing origins, Agile in knowledge-based industries must disassociate itself from building and production metaphors and attempt to perceive itself more in terms of innovation, products and services.

Methodological aspects aside, the sobering reality about Agile is that in spite of broad consensus that it enables higher levels of productivity and faster time to market, its adoption remains a challenge for most organisations. In particular, organisational and cultural barriers continue to hamper Agile within organisations as does lack of middle management support. Indeed, it is often these managers who express concerns about loss of control and have the greatest difficulty adapting to new practices (e.g., servant-leadership, people management and empowerment). Interestingly many surveys of practitioner and managerial staff reveal that few have substantial experience of Agile which suggests that support is required in order to ensure a successful transition. Although there are many training courses for operational staff (e.g., Agile PM Practitioner, DSDM Advanced Practitioner, Scrum Master and Product Owner) there is a dearth of suitable opportunities for managerial staff beyond introductory on-boarding. Thus many middle managers are faced with multiple challenges (e.g., how best to structure teams around self-organisation principles, building teams and resolving conflicts) and may lack coping strategies for relinquishing command-and-control mindsets. Accordingly, agile projects deliver mixed results when assessed by management with organisational integration issues and lack of support, experience and training appearing often at the top of the complaints list. The flip side is that when such matters are overcome, the results can be dramatic with few of those having successfully made the transition wishing to return to their pre-agile practices. In this sense, Agile can expose issues within the organisation (e.g., organisational barriers, stagnant management) that go some way to explaining poor performance in the marketplace (e.g., tardy delivery, reactive innovation).

To a degree some of the tools used in this chapter (e.g., agile charting, methodological dimensions) may assist in assessing the placement of a methodological approach (or a hybrid variant) in terms of the wider landscape. However, methodology is only a small part of the overall picture with the organisational and socio-pyschological aspects tending to take precedence. Many of the central themes of Agile (e.g., self-organisation, flexibility and autonomy) are revisited in detail in later chapters but for now it suffices to point out that given the relative cost of coordination and delay incurred through poor communication between distinct subsystems (e.g., business, solution development, testing and QA), a reduction towards tighter integration and greater collaboration is warranted. Such a transformation entails an inevitable amount of pain during which identities at the individual and corporate level must be reforged. These include a movement away from specialist individuals and silo-based structures, towards skill sets that comprise both technical and social aspects. The dynamic of the

marketplace and the central role that innovation continues to play therein also calls for a real commitment to organisational learning, a genuine culture of communication and collaboration and emotionally intelligent leadership and people management that redefine the notion of agile management as an enabler and facilitator of value delivery.

Part I
Agile Strategy

Strategic management is about explaining the performance of an organisation along with its ability to maintain that performance. Though there are several schools of thought concerning how performance arises within organisations, two particular approaches (the dynamic resource-based view and institutional theory) appear to be well suited to agile environments. Together, these enable the framing of agile as a strategic concern focused on the continual renewal of the organisation and its definition through a dialogue concerning the politics of meaning with its stakeholders. Thus the agile organisation is understood as a hub of innovation whose leaders must assume both the mantle of challenger and of entrepreneur. Rounding off these discussions of strategy is a brief analysis of the financial aspects of agile necessary in order to ascertain how it contributes to return on investments and ultimately the bottom line.

Chapter 2
Strategy and Innovation

Abstract When engaging in matters of strategic importance, agile organisations require an action oriented perspective that is based on market dynamism driven by the realities of the extended enterprise, globalisation, mass customisation, digitalisation and innovation. Accordingly, the classical strategic management approaches based on market and industrial analysis (e.g., marketing audits, SWOT) are found wanting due to their static outlook and lack of embracement of change and uncertainty. Instead agile strategy may be better framed in the language of dynamic capabilities and the negotiation of institutional legitimacy through a debate of the politics of meaning. Seen in this light coping with change and uncertainty requires the very organisational, emotional and intellectual basis that is so conducive to agile thinking and action. Within this context therefore the role of the agile manager is to be understood both as an entrepreneur of change and a challenger of the organisational status quo.

2.1 Introduction

There can be no doubt that in recent decades the rate of change in the marketplace and the industries that serve them has increased. This is commonly attributed to drivers such as the extended boundaries of the enterprise, mass customisation, globalisation, digitalisation (incl. Big Data) and innovation. Whilst there is clearly an increased rate of change, not all of it is attributable to these factors as evidenced by long standing practices such as planned obsolescence that exist to drive consumer spending. Extensions to enterprise boundaries refers to the formation of loose coalitions of networks of organisations that together provide products and services often with the objective of making their value chains more cost-effective, transparent, and responsive and are therefore referred to as value networks. Each participant may be involved in a different aspect of solution development and delivery implying that the production process may be fragmented and distributed. Generally speaking, however, the overall configuration depends not only on the nature of the process but also on market parameters (e.g., size and distribution of customer segments). Extended enterprises can be structured as formal supply chains or networks or may be bound in

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_2

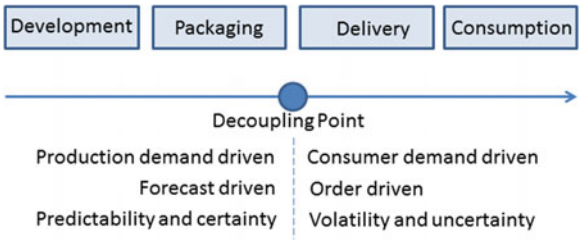

Fig. 2.1 Characteristics of the Decoupling Point of Products. Published with kind permission of © Alan Moran 2015

terms of partnerships or alliances as in the case of virtual enterprises.[1] Responsibility for individual phases of production (e.g., development, packaging, delivery and consumption) may be assumed by different participants often collaborating directly with each other and sharing insights and knowledge. This contrasts starkly to the traditional view of supply chain management and its adversarial relationships between suppliers.

When considered at the production level, a better insight into what is driving change can be gained by understanding at what point customers become involved in the solution development and delivery process which in turn helps clarify how the mechanisms of mass customisation come into play. As discussed in Chap. 1, the manufacturing industry already employs the notion of a decoupling point that marks the transition from lean and cost efficient mass production techniques to customised agile practices [283]. Alternatively, the decoupling point can be understood as the point in the solution delivery process that separates certainty and uncertainty of decision-making in relation to customer demand insofar as this affects production and distribution. Thus activities before the decoupling point are forecast and production demand driven, whereas activities after the decoupling point are order and customer demand driven. The movement of the decoupling point in the value chain of the extended enterprise away from customer therefore admits more uncertainty and variation in demand (see Fig. 2.1), a trend that has increasingly been observed in recent decades. This indicates a switch away from forecasting based planning (e.g., exemplified by full up-front specifications) towards more agile techniques that are better able to cope with the rising uncertainty.

The mechanisms for mass customisation are varied but include the modularisation of products and services during either development (e.g., reusable architectural components) or delivery (e.g., packaging based on service level packages) or the manner in which they are marketed or provided (e.g., SaaS[2]). Thus mass customisation is capable of delivering on customer's needs through more specific and tailored products and services. This may occur through the direct involvement of customers such as adherence to lean startup principles (i.e., collaboration), provision of a means for product or service configuration (i.e., adaptation), enabling visual alterations to

[1] Virtual enterprises owe their name to the use of ICT infrastructures to link organisational structures in a malleable and flexible manner reflecting the fluid and often transient nature of the association.

[2] SaaS refers to Software as a Service, an on-demand delivery paradigm for software assets. Similar concepts apply to Infrastructure (IaaS) and Platform (PaaS) provision.

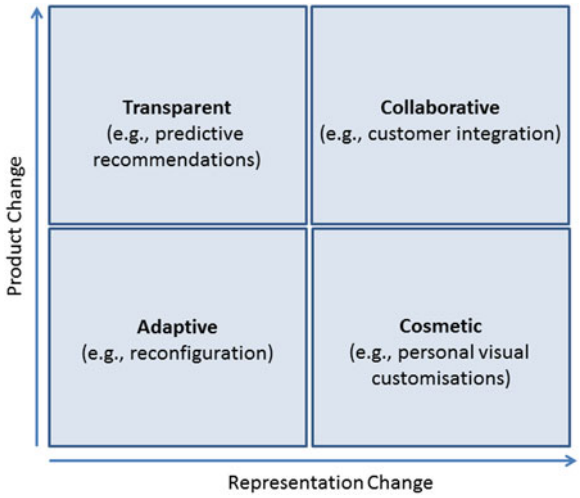

Fig. 2.2 Four Approaches to Mass Customisation. Adapted from [92]. Published with kind permission of © Alan Moran 2015

the representation of the product or service (i.e., cosmetic) or deriving offerings from combinations of existing standard packages (i.e., transparent) each of which reflects the extent to which the product or its representation has changed as depicted in Fig. 2.2 [92]. For example, the position of the decoupling point in the vicinity of solution development might suggest that modular components can be created with a view to enabling a simple form of adaptive customisation through the enablement and configuration of optional modules. This does not constitute a major customisation opportunity from the point of view of the customer, however, since it is likely that neither the product nor its representation (beyond the visibility of enabled components) have changed significantly. This same mechanism might, however, equally be used to entirely open up the internal architecture of the solution to enable a more collaborative form of customisation[3] where the customer together with the solution provider engage in creating new and highly tailored solutions in what has in effect become a transformation of a product into a platform.

The extended enterprise and mass customisation are the supplier and consumer sides of a global enactment of trade and economic activity. Whilst there is no universally accepted definition of globalisation, the term broadly refers to increased levels of economic activity arising from the trade and consequent flow of finances between countries, the diffusion of ideas, knowledge, information and technology and the changing nature of social, cultural and political forces that serve to erode national cultures and are linked to the growing influence of multinationals (incl. media and ICT providers). Thus globalisation serves as the arena connecting value networks, market participants and customers enabling a truly global distribution of development, packaging, delivery and consumption of products and services in an ever more

[3] In IT architectural models such as OSGi have transformed products into platforms atop of which customers can create their own specific solutions through open collaborative models (e.g., open source development).

flexible and tailored manner. The strategic response to this situation is to become hypercompetitive. Hypercompetition is fundamentally a strategic consideration arising from price-quality positioning strategies that lead to rapidly escalating competition aimed at capturing of the mindspace around new knowledge [60]. Hypercompetition exploits first-mover advantage built upon alliances between existing market participants or disruptive entrants to the market. Marked by frequent and audacious actions this leads to a situation of continual change wherein those enterprises without the appropriate capabilities will be found wanting and over time may be pushed entirely out of the market. Indeed, a good indicator of the fate of companies is an analysis of the 2003 Fortune 1000 institutions, seventy percent of which had disappeared by 2013 [148].

A key element of hypercompetition is the gathering of market intelligence and the processing thereof. The rise of social media and online usage has seen an explosion in the levels of data being captured and processed owing to the high levels of digitalisation of business. This in turn has transformed the manner in which solutions are developed, delivered, marketed and consumed resulting in a radically different dynamic not only between customer and suppliers (e.g., transparent price comparisons) but also between customer and customer (e.g., reviews and peer recommendations) leading to a loss of central control over the framing and direction of product and service development. There has also been fundamental shifts in the manner in which products are delivered (e.g., digital media) that affect the manner in which they may be consumed (e.g., license conditions tied to individual users) or transferred (e.g., restrictions on sharing of digital media assets or destruction of secondary markets). Digitalisation has therefore become a preferred channel in some markets. Furthermore, it may be supported by the gathering of data and market intelligence that has the effect of improving the targeting of marketing and optimising expenditure.

Within this context of hypercompetition fed by the information flood of digitalised business, product lifecylces are becoming increasingly shorter which in turn requires greater flexibility from solution teams in order to be able to cope with and adapt to the dynamic of changing business requirements and technology [38]. In fact, this flexibility has been identified as a key success factor in many projects particularly those concerned with product development and delivery [150]. Flexibility at the organisational level reflects proactive, adaptive and resilient abilities to adjust both the behaviours and the structures within the organisation. Flexibility therefore constitutes a competitive advantage that underpins innovative capabilities. This translates to adaptiveness (i.e., the ability to respond effectively and efficiently to change) at the team level in which autonomy and diversity play a significant role [97]. This impact is particularly observed in innovative environments where the ability to reframe problems through new perspective leading to creative solutions places demands not only on individuals but how they perform within groups (e.g., openness to the views of others). Innovation, closely tied with flexibility and creativity, is increasingly finding its muse in the ideas of ordinary individuals rather than elitist specialist functions [100] and is creating pressure on organisational structures to recognise this fact (e.g., high degrees of team level autonomy including veto rights on new hires, financial transparency and equity in terms of remuneration).

Thus the realm of globalisation in which extended enterprises use their mass customisation, digitalisation and innovative strategies to challenge the processes and supply chain management thinking of past generations has led to new disruptive forms of emergent strategy. Indeed, most projects could be considered to be sustaining in the sense that they represent a gradual and continuous improvement of knowledge and technologies. What characterises a discontinuity is the nature of the knowledge leap that enables entry into a new market based on the recognition of a need and the pace of development which often escapes wider market awareness. This is generally accompanied by a significant degree of risk and may lead to the creation of a new sphere or economic activity or the disruption of an existing market. What drives this process is the proximity of customers to those companies that possess the resources and agility to respond to their needs. For example, once a need is identified, Agile would suggest that customers (rather than managers) drive solution development, that the unit around which solution development is structured is independent of the overall organisation in which it is embedded (e.g., self-organised and autonomous) and that the size of the unit is linked to the size of the market.[4] Learning plays a central role in such organisations since often the target market might not even exist which necessitates a sense-and-adapt discovery mode rather than a planned execution approach. This includes explorative searching, iterative development and incremental delivery practices that use feedback to refine and adapt emergent solutions. Thus a widening of the technocratic focus of solution development is called for in order that customers are integrated early as drivers. In addition, marketing must become part of the deployment process in order to engage new (and perhaps as yet non-existent) markets. Fundamental to this dynamic are the culture and values of an organisation and the ability to sustain organisational learning to forge new capabilities [46].

These ideas are mirrored in the thinking of lean startups which promotes the notion that elimination of uncertainty can be established through feedback and learning infrastructures that facilitate explorative and iterative solution development rather than analytic strategic planning and forecasting [216]. This encourages double loop thinking wherein the basis on which solution development is founded is called into question (e.g., whether or not product or service should even be developed) which in turn requires customer involvement with whom experimentation and iterative development can be undertaken. Solution development must therefore address a specific need by creating a minimal usable solution (e.g., the Minimum Useable Subset of DSDM as described in Chap. 4). Faced with uncertainty, there remains only hypotheses that need to be tested in the form of validated learning together with the customer base (e.g. verification of the integrity, viability and sustainability of the endeavour).

[4] The "firm within a firm" is a common feature of some large organisation e.g., Johnson and Johnson comprises of over two hundred autonomous companies.

2.2 Strategic Management

Strategic management is about accounting for the performance of an organisation along with its ability to maintain that performance through any number of potential explanatory factors (e.g., innovation, capabilities, competencies). Accordingly it has both a descriptive (e.g., which factors and models best explain performance) as well as a prescriptive (e.g., what must an organisation do in order to perform) character. What constitutes performance, however, depends on the circumstances of the organisation and goes to the heart of why it exists (e.g., maximisation of shareholder value, optimal realisation of humanitarian or political ends). As a result, strategic management has come to encompass many facets ranging from the planning of an intended course of action, gaining an understanding of the environment in which an organisation operates, capturing the impulse to engage in different activities, grasping the underlying patterns of organisational behaviour and describing a specific perspective or outlook inherent in the organisation [178]. Strategy formation is typically a matter of identifying (e.g., mission setting and agenda setting), diagnosing (e.g., analysis of internal and external environments), conceiving (e.g., option generation and selection), realising (e.g., taking action and performance control). At the heart of this cycle is the tension between envisaging and planning for a possible future whilst accepting the uncertainty therein and the need to explore and adapt along the way [179]. Deliberate strategy represents thought followed by action as a result of which commitment can be secured, resources allocated and coordinated and enacted at the appropriate level (e.g., as a change programme). Emergent strategy, however, is an iterative form of thinking and doing wherein a coherent pattern of behaviour emerges over time that is characterised by opportunism, flexibility, learning and entrepreneurship guided by appropriate support. Thus it is seldom that strategy is entirely a planned activity wherein an intended course of action is determined and implemented leading to predefined benefits and outcomes. Rather, there is a significant element of emergent strategic influence that determines the ultimate course of direction and outcome.

There are several schools of thought regarding strategic management though many share a common foundation concerning the relationship between an organisation and its environment and how these translate into competitive advantage leading to performance. These theories can be broadly classified into economic and sociological camps though a sound approach to strategic management suggests that an appropriate combination of approaches is to be preferred over the reliance on a single theory or model. Economic approaches include market based thinking that is focused on needs, wants and values and how these may be satisfied by products and services, industrial organisational approach that popularised the notion of five forces that shape the marketplace leading to three generic strategies of competitive strategy as described in [207]. The resource-based view discussed in this chapter also can also be classified as en economic theory. Sociological theories of strategic management, on the other hand, include stakeholder and organisational politics theories along with the institutional theory approach discussed in detail later. Needless to say strategic man-

agement is no mere theoretical undertaking but one that requires analysis, foresight and continual validation and adaptation. Its findings can challenge the existence and nature of the organisation and it should therefore come as no surprise that in this context Agile has been referred to as disruptive [235].

Not all approaches to strategic management are appropriate for the agile organisation. Indeed, the weaknesses of the traditional marketing based strategies lie both in the assumption of an objectively knowable reality in relation to the business environment as well as the belief that customers have well established needs and wants.[5] Such approaches simply cannot cope with the ambiguity arising from excessive change or emergent patterns. For example, the classical industrial organisation theory, which is grounded in the patterns and competitive strategies of well established industrial groups, illustrates these limitations in respect of change and uncertainty preferring instead to describe a more static rather than dynamic view of industrial development [207]. There are, however, two specific lines of thought concerning strategic management that seem to fit the agile world particularly well in that they embrace change and uncertainty at their core. These include the economic dynamic resource-based view which conceptualises firms as bundlings of resources and capabilities that can be reconfigured to cope with the vagaries of the external environment and the sociological institutional theory approach that captures how interactions between individuals give rise to shared meanings and culture. Combining these two approaches provides a firm basis for formulating agile strategy based on an understanding of the sources of competitive advantage (that agile approaches offer) and how they can be translated into performance. The combination of the dynamic resource-based view and the institutional theory and organisational culture approach does not preclude involvement of other strategic approaches (e.g., stakeholder approach or agency theory) where appropriate. The focus of this chapter, however, is on strategic management thinking that best address the volatility and uncertainty that agile organisations claim to master and hence discussion is limited to these two central approaches.

2.2.1 Dynamic Resource-Based View

The resource-based view (RBV) school of strategic management focuses internally on the organisation and assumes that firms, understood as the bundlings of resources and capabilities, differ only in their possession of such bundles and that such inherent differences between firms persist for sustained periods of time [273]. This approach considers the tangible and intangible assets (e.g., expertise, technology, location, abilities) that could be reasonably associated with the organisation and assesses them to identify which possess attributes that are valuable, rare, inimitable and non-substitutable (collectively abbreviated as VRIN):

[5] Marketing based approaches to strategy formulation rely on performing a marketing audit (i.e., external analysis in terms of product markets and SWOT, examining product lifecyle and studying product-market and market growth-share strategies).

- *Valuable*. The ability of a resource to contribute to a competitive advantage (e.g., capacity to deliver a cost advantage). Value in the VRIN model should not be confused with the economic value of the resource which in a perfect market would be equal to its cost of acquisition and therefore not be capable of delivering competitive advantage.
- *Rare*. The scarcity of the resource assessed in terms of its ability to deliver a unique strategy to provide competitive advantage.
- *Inimitable*. The extent to which other organisations are inhibited from acquiring the resource. For example, a resource might not easily be imitated or reproduced or there might exist barriers to employing the resource (e.g., patents can create barriers that result in a resource becoming inimitable).
- *Non-substitutable*. Limitations on substituting the resource with an alternative that is capable of delivering the same outcome (i.e., resources of equivalent strategic value).

Not all resources that contribute to competitive advantage must possess all of these attributes. For example, core competencies are those resources that must simply exist for the organisation to participate in the marketplace (e.g., understanding of physics in the nano-technology field) and can hardly be considered rare. Definitions of resources and capabilities are described in [96] which also notes the subtle difficulties in identifying resources. Indeed, some managers expression a degree of frustration in not being able to identify precisely which dynamic capability is contributing to their success! An excellent overview of dynamic capabilities on which some of this material is based can be found in [77]. Resources may, however, be employed by dynamic capabilities which are defined as the organisational and strategic routines that managers use to change, adapt and integrate their resources in order to create new strategies. Examples of dynamic capabilities include:

- Product development routines (e.g., specific configurations of agile methodologies);
- Approaches to strategic decision-making (e.g., the ability to rally diverse pools of expertise to make decisions);
- Knowledge creation, sharing and transfer processes;
- Resource allocation routines particularly when they concern scare and valuable resources;
- Ability to create new synergies through the leveraging of collaboration within the organisation;
- Flexibility and adaptability in relation to the external environment (e.g., the matching of segments to changing customer needs); and
- Resource management (e.g., acquiring knowledge or expertise from outside the organisation or ridding the enterprise of resource configurations that no longer perform).

VRIN resources form the basis of sustained competitive advantage when engaged by dynamic capabilities since they can be continually developed in a manner that is

hard for competitors to reproduce. Underpinning dynamic capabilities are the organisational routines, known as core capabilities, built upon VRIN resources that are repeatedly brought into play as part of the dynamic of strategy making (e.g., specific combinations of nano-technological methodologies, infrastructure and people). Indeed, it is argued that performance in the marketplace is down to imperfections in competition arising from differences in the possession of such critical resources and their deployment in emergent strategies. Thus resources do not alone explain competitive advantage in highly unpredictable and changeable environments but rather it is their interplay with capabilities, wherein the role of knowledge management plays a crucial role, that is of interest.

IBM is a firm that possesses nano-technological skills ordinarily found only operating in the computing industry but which might create new opportunities elsewhere by reconfiguring its resources (including the acquisition of new expertise and abilities) to enter new markets which in turn might rely on how it acquires the necessary knowledge and the process it uses to manage its internal resources.

For example, employing nano-technology in the medical field serves as a good example in this respect [243]. Drug resistance is on the rise spurred on in part by the manner in which patients are treated. For example, interruptions in the treatment of tuberculosis, which has seen a steady rise in the Asia Pacific region in the past, can result in strains mutating, thereby becoming resistant to more stronger forms of treatment. Some traditional treatments have undesirable side effects (e.g., one popular antibiotic carries a risk of fatal heart arrhythmias particularly in individuals with magnesium or potassium deficiencies) which requires care and attention when administering drugs. IBM, a company long associated with corporate IT solutions, has since 2011 been researching the application of synthetic biodegradable nanoparticles to attack bacterial cells that are resistant to antibiotics. This claims to reduce a variety of other risks associated with the treatment of patients including better protection of medical staff. The technology, which targets bacterial membranes by breaking into cells, adopts a fundamental different strategy to traditional medicines and can be administered as a gel (e.g., passed through a tracheal tube) or as a soap to tackle surface wounds. From a strategic perspective this illustrates the interplay between core capabilities (e.g., nano technology methodologies and skills) and dynamic capabilities (e.g., marshalling of technical and medical knowledge) to create new strategies leading to competitive advantage (i.e., superior treatment over conventional medicines).

The performance logic of the dynamic resource-based view considers coping with volatility as the central tenet of its survival process in which the development of core competencies plays a vital role and the strategist is envisioned as an entrepreneur [245]. Thus the role of the organisation is to be the custodian and promoter of

dynamic capabilities in a volatile environment in which exploration leads to future competitive advantage that delivers superior organisational performance. Since the exact nature in which resources are developed and lead to a positive contribution to overall performance, however, is often difficult to trace (let alone predict) the dynamic resource-based view approach is often claimed to have an explanatory rather than a prescriptive character. Not content with only meeting market needs, this strategic approach aims to extend existing boundaries wherein creativity, ideas and persistence pay off in new combinations of resources that give rise to new patterns of economic activity. Strategic engagement in the resource-based view revolves around the experimentation with new alternatives thus making clear the link between strategic management and organisational learning [160]. In this context the need for learning (incl. open exchange across the value network), a culture of tolerance towards mistakes and adequate time to absorb and reflect on learning all feature prominently.

Thus the dynamic resource-based view sees capabilities as arising from within the organisation and exhibiting traits that are not readily transferrable to other organisations (e.g., entrepreneurial spirit, corporate culture) and have the capacity to cope with market volatility. Interestingly, dynamic capabilities in stable environments often resemble the traditional routines found in ordinary organisations (i.e., detailed knowledge based on processes with predictable outcomes). However, in highly dynamic environments these capabilities assume less stable characteristics which rely on new knowledge (including feedback loops) and iterative approaches leading to adaptive outcomes. What emerges is a picture in which it is not the dynamic capabilities, which over time assume the mantle of best practices, but rather the unique resource configurations that they produce which is the source of competitive advantage. In fact, an agile methodology considered as a resource, whilst surely being valuable is rather unlikely to be rare, inimitable or non-substitutable since it is very conceivable that broadly similar outcomes are arrivable at through many routes (e.g., the use of any one of a plethora of agile methodologies) based on broad commonalities (e.g., agile principles).

Application of the dynamic resource-based view approach to agile environments suggests that the capabilities found in an agile development process could be described as best practices since it is likely that different organisations who adopt a specific methodology (e.g., DSDM) will broadly adhere to similar practices. However, whilst commonalities are clearly present, it is in the detail of implementation that differences are observed. This may include the ability to form cross-functional teams, the capacity to share knowledge and create common understandings (e.g., manner in which facilitated workshops or retrospectives are conducted) or the integration of outside sources of knowledge (e.g., access to Business or Technical Advisors) and influences (e.g., balance of internal and external autonomy-see Chap. 11 for a description of the three levels of autonomy and their impact on agile team performance) [128]. Thus any commonality that organisations share (e.g., adoption of DSDM) does not imply that the dynamic capability they derive is the same. For example, organisational learning has a key influence on how Agile develops within the organisation and is played out on many levels (e.g., project team, interaction between business units, access to internal and external thought leaders) or may take many

different forms (e.g., social bonds, formal alliances or partnerships). This means that management of dynamic capabilities lies with the agile manager who must consider how agile is being deployed and what idiosyncrasies of the organisation contribute to or detract from its effectiveness. Once identified, a dynamic capability requires practice in to order to accumulate the tacit knowledge required to sustain it. Being Agile, therefore, clearly means more than just blindly following a method.

It would seem that even in industries that develop slowly there is still a need for dynamic capabilities though these rely more on existing bases of knowledge that may already be codified into standardised processes. More dynamic markets in which boundaries are blurred or ambiguous employ dynamic capabilities which feature knowledge creation and "unlearning" practices [256]. Routines in such environments are guided less by detailed rules and more by general principles that define boundaries of activity and behaviour. For example, when forging alliances Yahoo used two simple rules to govern deal-making [78] and during the 1990s Intel used a single rule for wafer production [37]. This level of detail is sufficient for decision-making which thrives in an atmosphere of openness and creativity where guiding visions drive solution development. The role of communication and collaboration is evident in the experimental and iterative approaches (e.g., prototyping) endowed with tight feedback loops often found in such environments which have a striking resemblance to successful agile environments.

Dynamic capabilities endear team flexibility which turns out to have an important impact on solution quality. In this context flexibility refers primarily to adaptation and responsiveness towards the external environment and presumes an appropriate degree of matching between the project team and its business environment. Flexibility is evident both in the time and effort it takes to adapt to change as well as the scope of change with which a team or organisation can cope with [93]. Sources of flexibility can be found in the psychological composition of the team, the nature of organisational structures, cross fertilisation of ideas, customer integration, the characteristics of IT architectures and the tooling employed by the enterprise. In particular the flexibility of teams can be understood in terms of response extensiveness (i.e., the range and variety of socio-technical changes that the team is capable of responding to) and response efficiency (i.e., the effort required by the team to manage business and technical change) as outlined in [152]. These in turn are linked to two specific dynamic capabilities, namely the reactive ability to cope with changes late in the project which can be nurtured by integrated tooling and prototyping practices and anticipatory abilities to proactively tackle ambiguity and potential changes early on in the project which requires collaboration, involvement and proactive risk management [152, 263].

Cultivating these capabilities requires management of project uncertainties (e.g., application of agile risk management techniques), continual observance of technological trends, an environment of inclusive and proactive thinking fostered by communication, collaboration and coordination. This captures both anticipatory and reactive elements that use resources such as integrated tool environments and highly skilled team members. Indeed, research indicates that reactive capabilities have a marked effect on response extensiveness and efficiency whereas anticipatory

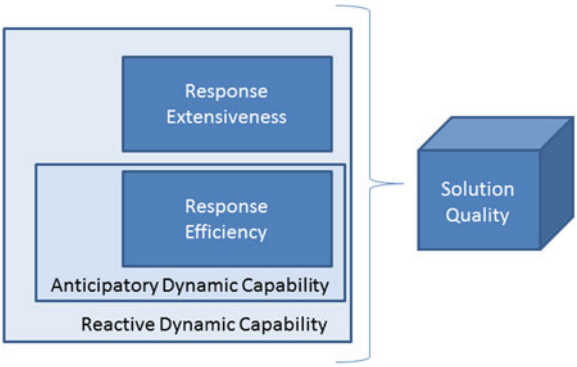

Fig. 2.3 Influence of Anticipatory and Reactive Dynamic Capabilities on Solution Quality. Based on [152]. Published with kind permission of © Alan Moran 2015

capabilities influence efficiency indicating that possession of these capabilities enhances team flexibility (see Fig. 2.3) [152].

Moreover, it would appear that this flexibility has a significant bearing on solution quality. To appreciate this point it is important to understand the importance of flexibility in solution design (i.e., its ability to accommodate changing business requirements and needs) in mitigating the potentially detrimental effects of change on solution quality. Therefore, team flexibility is central to ensuring that a solution continues to retain its fitness for purpose and fitness for use in spite of changing circumstances which in turn can only be ensured by flexible architectures (an anticipatory consideration) and integrated tooling (a reactive consideration). These alone, however, do not suffice since their interaction with other factors (e.g., people skills, communication and collaboration, architecture investment leading to reduction of technical debt and involvement of appropriate expertise) all contribute to the final outcome [123, 157, 162, 262].

2.2.2 Institutional Theory

This approach to strategic management shares some of the aspects of dynamic resource-based view albeit from a sociological point of view. In this approach, institutions are considered vassals of economic activity and a manifestation of social order that govern how individuals interact with each other (e.g., by defining normative systems and cognitive understandings that give meaning to social exchanges often in terms of activity). At heart is the belief that regular interactions give rise to shared meanings and culture. This chimes with the ethos of attributing to a vision, meaning born of experience, communication and collaboration as found in agile communities.

Institutions embody habits and routines which over time lay the foundations for standards, processes and procedures within the organisation. Yet the existence of informal networks serves to undermine the notion of a rational organisation optimised around structures and well-defined practices. Informal networks of an organisation are often masked by more formal structures in an apparent gesture of conformity that

endows the organisation with legitimacy by adhering to socially constructed norms, values and beliefs [67, 173]. In other words, in appearing to be a more rational and structured entity rather than a bazaar, a firm hopes to appeal to wider expectations of itself in terms of how it wishes to be perceived and be seen to act. Indeed, this might explain in part some of the resistance towards fluid organisational structures found in some organisations attempting the transition to Agile. However, it is often precisely such networks that enable organisation to adapt to changes that more formal and rigid structures might otherwise struggle with. Thus the process of institutionalisation assumes strategic relevance since it influences the overall performance of an organisation. What is called for in innovative and agile organisations is a culture that challenges and calls into question the traditional ways of doing things together with the status quo. Thus the process logic of the institution theory approach to strategic management involves the confrontation of institutional pressures by engaging in the politics of meaning wherein the strategist plays the role of the challenger.

At the root of performance in the institutional theory approach is the notion that legitimacy underpins performance. This is due to the fact that the establishment of new institutional norms requires legitimacy and the consent of stakeholders which necessitates a closer examination of what constitutes legitimacy. Legitimacy can be defined as the generalised assumption that the actions of an entity are desirable, proper and appropriate in respect of some socially constructed system of norms, values and beliefs [252]. This means that legitimacy has collective and social nuances which are important since it is needed to maintaining credibility and soliciting support both of which are performance drivers. However, legitimacy assumes different forms and the best known classification of these is in terms of pragmatic, moral and cognitive legitimacy:

- *Pragmatic legitimacy*. Self-interest of the organisation and its immediate constituency who in turn may both scrutinise the organisation to determine the consequences of its behaviour for them and support its policies based on its expected value to them.
- *Moral legitimacy*. Evaluation of the organisation in terms of social norms and beliefs about what the right thing to do is. Moral legitimacy does not preclude an assessment wherein the evaluator conflates personal and societal benefits to cynically manipulate the situation. That aside, however, the premise of moral legitimacy maintains that moral concerns trump self-interest. This evaluation may be based on the accomplishments of an organisation, by embracing socially accepted techniques or practices, the capacity to undertake specific kinds of work or to a limited extent the personal charisma of an individual [252].
- *Cognitive legitimacy*. Assessment of the organisation based on comprehensibility (e.g., offering a plausible explanation for the existence of an organisation that meshes with wider belief systems and the experiences of those affected) or inevitability and permanence (i.e., considered so established that the alternatives are unthinkable).

The logic behind the institutional theory and organisational culture approach is therefore to confront the institution by engaging in politics and meaning wherein the

role of the strategist is the challenger. This in turns means that the organisation is cast in terms of its ability to mobilise skills using leverage points found within its environment in order to affect a transformation that delivers the necessary legitimacy that underpins performance. Thus the strategist must identify leverage points within the organisation and assess their mobility and capacity to bring about institutional change without excessive loss of meaning. Legitimacy is important in this approach since a shared meaning requires a common language and understanding of the situation (i.e., cognitive legitimacy), shared norms and values (i.e., moral legitimacy) and claims on the resources of the organisation with which to implement strategy (i.e., pragmatic legitimacy). This means that change ushered into an organisation must take account of the current and future states of the organisation as well as political and cultural considerations.

2.2.3 Agile Implications

Linking the institutional theory approach back to the dynamic resource-based view, the reliance on anticipatory and reactive capabilities that underpin team flexibility suggests that the role of the agile project manager becomes one of negotiating new meaning within the project in the light of emergent uncertainty thereby becoming a guide for the long term solution vision. Thus both approaches to strategic management (dynamic resources-based view and institutional theory) find common ground in the nature of decision-making and organisational design in which strategy is articulated in rationalist terms. Therefore strategic management and agile share key elements common to both disciplines. For example, the capacity to cope with change and uncertainty through an appropriate mixture of planned and creative elements that ultimately determines an emergent course of action is central to both as is the need to engage in political debate (concerning meaning) with all stakeholders and to incorporate therein periods of reflection, adaptation and feedback. Plan centric thinking on the other hand must contend with the fact that they are at their most effective within an environment of certainty. Plan-driven approaches become the limiting factor where uncertainty and change prevail at which point adaptive and multi-tiered planning (e.g., coarse initial planning over the long term, followed later by detailed short term planning) becomes more appropriate.

The organisational perspective on strategy envisages a situation in which organisational culture has established a shared set of assumptions governing the tackling of problems which has sufficient validity to be transmitted onto new members to guide their perception, thinking and action. Thus when an issue arises it may solicit a response of no action (i.e., ignore the issue), be greeted as something consistent with the beliefs with which the organisation identifies (i.e., be treated with known and familiar patterns) or present itself as something at odds with the organisation and is difficult to interpret [135]. The last of these cases attracts either a superficial response or initiates a political struggle to resolve the ambiguity within the parameters of the existing paradigm. This is precisely the conformist attitude towards institutional pres-

sures that results in a convergence of similar types of organisations (e.g., that which makes all banks similar). Inverting this picture is entrepreneurship (or its institutional variant, intrapreneurship) which understands its role as confronting institutional pressures by engaging in the politics of meaning in which strategy poses a challenge to existing views within the organisation [202]. This occurs through iterative reassessment and reinterpretation of held beliefs, often spurned on by innovation though sometimes necessitated by crises, that call into question the status quo. This invites conflict since the organisation must regroup and re-institutionalise itself around new a meaning.

The implications for agile strategic management lie in the recognition that institutional theory has a role to play when an organisation is confronted with the prospect of change (e.g., agile transformation) that threatens held-beliefs, frames of reference or claims on resources. This means that change will typically invite open conflict that revolves around a number of points linked to different forms of legitimacy with which the organisation justifies its existence. Thus an agile transformer must be capable of identifying and classifying those points that are the subject of dispute and formulate appropriate strategies for tackling them. Consider, for example, attempts to establish self-organisation within the organisation (see Chap. 10). This may give rise to issues relating to the balance of autonomy within the organisation which may challenge existing control and power structures since self-organised teams thrive in situations where there is low external and individual but high internal autonomy. In addition there may be discussions surrounding values such as trust and the nature of relationships between managers and their direct subordinates. Finally the fluidity and dynamism may question the definition of what a project team is in the self-organising context. Clearly what is being observed here is the emergence of concerns linked to self-interests, values and cognitive frames of reference (i.e., the three forms of legitimacy) even if the real sources of concern are being kept hidden or masked in other terms. Treatment of such disputed points often requires differing strategies that take note of both the central issues and how they relate to other matters. Since not everything is under dispute there also exists the opportunity to leverage undisputed points to help mobilise the disputed ones. This requires a careful analysis of all points to uncover the inter-relationships between them in order to plot a course of action towards a desired settlement. Finally, since it is seldom the case that a controversy arises in isolation, an agile transformer must be at least aware of the possibility of multiple over-lapping disputes that interact with each other, further complicating the process of determining a resolution that maintains overall legitimacy. The process of change therefore requires a renegotiation of meaning within the organisation that in time will itself become institutionalised giving rise to a new legitimacy.

2.3 Innovation and Entrepreneurship

Innovation revolves around behavioural and social processes with the goal of achieving desirable changes or avoiding the penalty of inaction [218]. To some, innovation is a form of creative destruction that must arise from within an organisation if it is to survive in its wider environment and it is this view of the entrepreneur that features in the dynamic resource-based view of strategic management [231]. Thus if it is the entrepreneur who initiates the creative value generating process, then it is the resources of the organisation that make possible its full realisation. This requires not only flexibility on the part of the entrepreneur (e.g., psychological mindset, openness to ideas) but also receptiveness and support (incl. skills and resources) on the part of the organisation itself.

Multiple studies have shown that flexibility, the ability to change, is integral to innovation though it manifests itself in different forms such as the ability to solve problems when old methods fail (e.g., by reframing the problem from another perspective thereby admitting a new solution) or the capacity to spontaneously create new responses without the need for external pressure [90]. Thus flexibility feeds the reorganisation of knowledge as well as the highlighting of connections between seemingly unrelated concepts that is the driving force of creativity. From a personality point of view flexibility is expressed by a preference for change and novelty as indicated by the Openness to Experience personality factor (see Chap. 11). Interestingly such flexibility may be found even at the level of personality suggesting that such individuals are capable of embodying seemingly different and incompatible traits whilst themselves remaining creative [79, 80]. Such individuals, who may have a wide range of interests, ought to be harnessed in change programmes within an organisation in which they often play an instrumental role.

Individual flexibility, however, is insufficient if the organisation itself is not receptive to new ideas and therefore managers not only play a key role in the assessment of ideas and their conversion into practice but are also key barriers to successful adoption. Organisational structure may influence the extent to which managers are willing to risk entertaining new ideas particularly where there is a perception of formal rigidity, approval processes and organisational inertia. Agile organisations exhibiting characteristics of fluid organisational structures with less hierarchical structures that are based on servant leadership principles together with continual communication and collaboration are far more likely to be in a position to foster and facilitate innovation. Practice has shown that permitting diverse specialists to work together transforms a group of otherwise rigid individuals into a flexible team [82]. Whether or not teams are congregated with this specific objective in mind or become diverse owing to agile practices, the end result is the same: exposure to different perspectives encourages flexibility thinking. Furthermore, individual or group ownership of problems (which includes identification with the task at hand) is known to motivate initiative and encourages a flexible search for solutions. Flexibility of action is thus another path to cognitive flexibility and openness towards the ideas of others and requires permitting individuals the necessary autonomy with which to work.

Entrepreneurship is associated with the creation of new ideas or modes of operation that is often attributed to vision, motivation, networking (incl. access to resources) and a willingness to take calculated risks. The precise model of entrepreneurship may vary and can be based on a vision of future state leaving open the precise route to the goal (i.e., causation), an inward focus on one's own abilities and knowledge (i.e., effectuation) or a continual process of experimentation to refine and develop ideas (i.e., experimentation). Entrepreneurship is linked to innovation in the sense that the former sets the context wherein the latter can be realised. There are pronounced similarities with entrepreneurial behaviour and attitudes towards flexibility discussed earlier in terms of the reframing of problems to discover new solutions, the tolerance of failure when seen as a learning exercise and the openness to new experience as a source of inspiration.

Despite popular perception, there is no single unifying definition of what an entrepreneur is, though there may be broad consensus that they come from diverse backgrounds, are often committed team players that exhibit high degrees of achievement and persistence and whose interests may be economically or socially driven. Moreover, their behavioural characteristics reflect creative and flexible approaches, and a continual learning attitude evident in an ability to cope well with change and uncertainty. Entrepreneurs may assume the strategic mantle of challenger of established norms (as described in the institutional theory of strategic management) helping an organisation change, adapt and evolve in volatile markets. In an attempt to alleviate some of the tensions that arise from intraprenuerial[6] behaviours, it is not uncommon to see organisations adopt "firm within a firm" structures that delegates resources and autonomy to sub-units enabling them to pursue their goals to a greater or lesser extent independently of the wider organisational context. Thus entrepreneurship thrives in an environment that is characterised by decentralised decision-making, a commitment to organisational learning and a tolerance of failure exemplified by courage and determination in the face of change.

To act on entrepreneurial urges, an individual must be capable of envisaging a future desired state and feel empowered to realise that goal (e.g., have the necessary support and access to resources). Desire without empowerment leads to frustration and empowerment without the vision usually implies diffusion of efforts. Yet despite this there remains the issues of alignment (i.e., ensuring that the entrepreneurial aspirations of an individual are in the spirit and purpose of the wide organisation) and tensions in management behaviour (e.g., the conflict that arises from the promoter entrepreneurial mindset and the trustee administrative function protective of the status quo). The former is as much the nature of the debate on the politics of meaning and indeed a certain creative tension is both desirable and unavoidable. The latter captures a fundamental dichotomy of entrepreneurship, in which the entrepreneur is identified as a promoter (see Table 2.1), that has been the topic of much discussion ever since the term entrepreneur was invented.

[6] Intraprenuership refers to entrepreneurial behaviours within established organisations i.e., those that are not considered start-ups.

Table 2.1 Characteristics of Promoter versus Trustee. Adapted from [250]

Promoter	Trustee
Driven by opportunities for which the necessary resources must be acquired and from which business strategy is to be determined in an emergent fashion	Custodian of resources that in effect limit opportunities that can be pursued and which determine the business strategy
Consumers of resources that are committed incrementally based on staged buy-in, controlled risk and growing trust	Up-front planning and commitment of resources that are ownership ring-fenced
Growth-driven thinking	Survival-driven thinking
Idea rich (often in excess of available resources) drawing upon sense-and-adapt experiences of the wider environment	Resource-rich (at the expense of ideas to exploit them) detached from societal developments
Adaptive and informal controls and management style based on flat and fluid organisational structures and largely horizontal flows of informational	Preference for standardised management principles and practices with controlling of resources and funds and commitment to plans, functions and work schedules based on vertical flows of information
Reward and status based on value generated with perhaps group based incentive schemes	Reward based on individual responsibility and position within the hierarchy

Agile thinking is in many respects compatible with the entrepreneurial view of the world. Both understand the need for a collaborative environment of decision-making, a culture that is tolerant of failure (in the name of experimentation) and neither are afraid to embrace change but rather both thrive on it. Those organisations that best fit the description of adaptive and entrepreneurial share a belief in their ability to influence the environment as well as a focus on desired future states characterised by growth or change and have the capacity to institutionalise these views into the fabric of the enterprise.

2.4 Management Implications

Rather than perceiving market turbulence as a threat, agile organisations are in possession of the necessary dynamic capabilities and the capacity to engage in politics of meaning with their customers and other stakeholders to exploit change as an opportunity. To quote an old Chinese proverb: when the wind of change blows, some people build walls whilst others build windmills. An agile perspective on strategic management suggests that it is not so much the practice of Agile but the specific manner in which it is exercised (e.g., team building process, ability to integrate customers) that gives rise to dynamic capabilities that deliver competitive advantage. The implications for management suggest that development of anticipatory and reactive dynamic capabilities foster team flexibility since the continual integration and reconfiguration of resources both inside and outside of the team improves its ability to effectively respond to change throughout the development process. Furthermore each capability

comes into play at different stages of solution development. For example, reactive capabilities are more important later in the process where investments in modular architectures and integrated tooling pay off. On the other hand, anticipatory capabilities are very project specific and rely heavily on learning, stakeholder involvement and proactive behaviours (e.g., agile risk management) which should occur early on in the project. The limits of anticipatory capabilities can perhaps be explained by the fact that they generally focus on critical elements (e.g., concerns that arise in risk identification workshops) rather than attempt to provide comprehensive forecasting. Be that as it may, anticipatory capabilities affords late decision-making by which time more accurate information may become available with which to manage issues. The important point to bear in mind is the interdependence of anticipatory and reactive capabilities which in unison give rise to team flexibility. Furthermore a state of continual change requires a continuous readiness to engage in political debate both within the organisation and with its stakeholders which requires a willingness to confront and embrace controversy and to negotiate new legitimacy and meaning for the organisation. The key facets of an agile manager are those of the entrepreneur and challenger. Thus Agile alone, though no silver bullet, does embody the correct spirit and approach with which to cope with change and drive forward innovation.

The suggestion that it takes a crisis to precipitate innovation is usually a reflection of an uneven distribution of power and decision-making within an organisation that causes it to only react once a tipping point has been reached. The alternative, an energetic and continually self-renewing organisation, requires no sense of crisis in order to engage in innovation. This is because the mindset and structure of the organisation exhibits more balanced distribution of power married with an internal flexibility that is capable of accommodating new initiatives on a continual basis. Indeed, flexibility, a cornerstone to promoting innovation, requires a shift away from the elitist opinion that good ideas must arise from specialist functions within the organisation (e.g., R&D or marketing departments) instead prompting efforts to leverage the entire organisation in the search for solutions to real problems. Thus a manager must seek to promote problem-solving of their employees by coaching and supporting them in the search for solutions (e.g., provision of resources). Managers can be assisted in this manner by broadening their own horizons through direct experience within their teams, flexible modes of action that promote openness, and relinquishing the sense of fear that may be felt when employees put ideas forward (e.g., some managers feel that it is their prerogative to come up with the ideas and therefore feel undermined when their employees come forward with suggestions). Thus managers may need to question the role that their organisation (including structures that they themselves have erected) have in stimulating or stifling flexibility and innovation.

More often than not there is a focus on innovation as a new product or service development model and less on the management of innovation. To achieve this there must be an ingrained sense of commitment to improving the quality of management within the organisation and this often occurs through the adaptation of (the more controlling) management processes (e.g., capital budgeting, project management, people and performance management). For example, being clear about what constitutes innovation in the context of the organisation and its markets, supporting

innovation through resources (e.g., time, budget), training and mentoring managers and staff in innovation, incubating, tracking and reviewing innovative projects are all manners in which management can assist innovation and its translation into bottom line performance. One approach to tackling management innovation is to focus on big problems (i.e., those with the greatest scope for innovative thinking), actively searching out new principles rather than attempting to apply existing ones (e.g., by leveraging diversity, distributing decision-making and soliciting passion and ingenuity), deconstructing existing management orthodoxies (e.g., by challenging them with alternate views) and exploiting the power of analogies (e.g., autonomy, self-direction, servant leadership) [100].

Chapter 3
Financial Management

Abstract In spite of the economic claims made for Agile (e.g., enhanced return on investment, cost risk management), financial and accounting aspects of Agile are a relatively neglected topic (e.g., advice on accounting for capital and operational expenditure, pricing models). Yet value delivery and economic risk mitigation lie at the heart of Agile one of whose central tenet is the use of feedback loops to validate and refine solutions. Moreover, the iterative and incremental structure of agile projects directly contributes to improved rates of return owing to the frequency of benefits enablement which improves the net present value of expected future cash flows. It is therefore prudent to embed classical appraisal thinking into agile projects (e.g., increment level assessment of return on investment) in order to demonstrate value to the customer and to ensure that an appropriately agile contracting framework and pricing model are in place that recognise the unique features of Agile whilst permiting the learning that takes place within such environments to enhance value.

3.1 Introduction

Projects deliver value within an organisation [15] and for projects whose value is defined in terms of revenue generation, project appraisal techniques play an important role. Not all projects, however, are intended to be revenue generation vehicles. For example, in the public sector projects may aim to deliver social benefits and in this context cost, rather than revenue, may be the primary driver. Such assessments usually take place early on in the project (e.g., DSDM suggests that cost-benefit analysis be done in the Foundations phase as described in Chap. 4) and constitute part of project selection at the portfolio and programme levels. There is, however, an inherent uncertainty in such practices since they are based on forecasting future cash flows the purpose of which is to compare the impact of expected future revenues against known costs. To an extent Agile alleviates some of this risk through incremental delivery which permits focus on shorter term certainties (i.e., where sufficient information is

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_3

already available on which to base decisions) but otherwise appraisal techniques are the same as for traditional projects, the most common of which include:

- *Payback Period*. This is defined as the period of time a project will take to return funds invested in it. This measure is limited since it takes no account of future profitability of the project beyond the payback period.
- *Net Present Value*. These factor in time value of money to reduce all future expected cash flows to a single point and use this calculation to determine if the project is viable (i.e., if the net present value is greater than zero). This is considered one of the most robust approaches to investment analysis but is subject to the uncertainty of forecasting future cash flows.
- *Internal Rate of Return*. The Internal Rate of Return (IRR) is that effective compounded rate of return which makes zero the sum of all future expected cash flows, discounted to present day. The computation of the IRR can involve some complicated algebra that yields a unique solution only in certain circumstances in order for it to make sense (e.g., a project with an initial capital investment followed by subsequent cash flows relating to costs and revenues but no sizeable decommissioning costs). Projects are deemed acceptable if the IRR is unique and is higher than the required rate of return reflecting the opportunity cost[1] of the funds.
- *Profitability Index*. This accounting based index is defined as the net present value of cash flows following the initial investment divided by the initial investment. Projects are deemed acceptable if this figure exceeds one.

Considered in terms of investments, the return on investment metric ought also to be used to assess emergent returns at regular intervals throughout the project (e.g., on delivery of an increment). Formally, return on investment is defined as the difference between the gains and costs of an investment divided by the investment amount (though there are some minor variations on this formula). The challenge with investments (especially in the IT sector [214]) is the determination of derived benefit since often projects result in evolutionary change that translates into moving targets from an appraisal point of view. Moreover sometimes the purpose of an IT investment is either to reduce costs (referred to as cost displacement or avoidance) or to improve the competitive advantage of business (e.g., creating a decision-making capability based on data analytics) affecting the manner in which value is assessed and evaluated. In general, however, in the commercial sector there is little point in IT investment unless this results in improvements in how business is conducted that lead to improved profits or returns on investment. Thus IT investment must be understood in terms of the business processes and practices they support and this necessitates a partnership of IT and business that is capable of agreeing what the final outcome of a project should be and how benefits are to be measured (e.g., formulation of a Business Case linked to Benefits Assessment as described in Chap. 4). Such benefits may be related to the efficiency of a business process expressed in cost or pricing terms (e.g., more output for the same level of input), improved solution quality or

[1] The opportunity cost is that amount which could reasonably have been expected to be made had the funds not been used for the project.

simply higher morale amongst the workforce. Therefore, whilst an economic focus prevails in this chapter, in practice any number of different metrics may be used to assess benefits and value generation.

3.2 Beyond Budgeting

Parallel to developments in the agile community, there arose during the 1990s a set of accounting based principles and practices collectively referred to as Beyond Budgeting [121] that challenged traditional performance management based on budgetary controls (e.g., formulation of mission statements, strategic plans and programmes and variance analysis). Central to this criticism of the budget process is that it is too expensive and cumbersome and is not appropriate for dynamic and competitive environments. Instead, Beyond Budgeting promotes techniques based on integrated shared goals and values (e.g., Balanced Scorecard) and activity based accounting. Intended to cope with fast changing environments this approach shares much in common with Agile and its own set of principles can be classified as follows in leadership and process terms:

- *Leadership Principles*
  - *Customers*. Focus everyone on improving customer outcomes, not on hierarchical relationships.
  - *Organisation*. Organise as a network of lean accountable teams, not around centralised functions.
  - *Responsibility*. Enable everyone to act and think like a leader, not merely following the plan.
  - *Autonomy*. Give teams the freedom and capacity to act, do no micro-manage them.
  - *Values*. Govern through a few clear values, goals and boundaries, not detailed rules and budgets.
  - *Transparency*. Promote open information for self-management, do not restrict it hierarchically
- *Process Principles*
  - *Goals*. Set relative goals for continuous improvement, do not negotiate fixed performance contracts.
  - *Rewards*. Reward shared success based on relative performance, not on meeting fixed targets.
  - *Planning*. Make planning a continuous and inclusive process, not a top down annual event.
  - *Controls*. Base controls on relative indicators and trends, not variances against a plan.
  - *Resources*. Make resources available as needed, not through annual budget allocations.

– *Coordination*. Coordinate interactions dynamically not through annual planning cycles.

The connection with Agile becomes clear when one considers organisational paradigm of self-organisation built around, entrepreneurial and empowered teams, customer involvement, autonomy and delegated decision-making including the freedom of the team to select its own quality and performance metrics, solution development methodology and control of funding (e.g., training, equipment). Moreover there is a clear rejection in the Beyond Budgeting movement of planning over long timeframes where insufficient information exists with which to make key decisions (e.g., setting of annual budgets) in favour of more short-term adaptive controls. For example, the annual budget cycle can have a debilitating effect on adaptive planning as the focus shifts towards the tracking of deviations from a plan and away from responsiveness towards customers needs. Crude measures such as the inclusion of a budget buffer to accommodate unanticipated events do little to resolve this issue owing to their fundamental misunderstanding of the underlying dynamics of change.

Performance management, considered here as a subdiscipline of management accountancy, is concerned with the structure of incentive schemes and the evaluation of performance. From an agile perspective, group rather than individual rewards are preferred since the latter tends to distort behaviours within the group that act against the interests of the organisation (e.g., non-sharing of information). Instead schemes based on profit sharing or relative performance evaluation tend to find more favour though their effectiveness does in part rely on positive group dynamics and conflict resolution (e.g., tackling of social loafing). Use of benchmarks, key performance indicators or subjective peer reviews are common in this context. In situations where teams are not competing for the same internal resources there can also arise positive interactions and knowledge sharing at the intra-team level that help improve coordination across the enterprise thereby enabling integrated performance management at higher levels.

3.3 Expenditure and Profit

Since profitability is a central concern of organisations it is necessary to understand what profit actually is. At its simplest level profit refers to revenues less the operating costs (e.g., salaries, facilities, expenses) incurred in acquiring them. However, the accounting of costs within an organisation also makes accommodation for long term liabilities that impact how profit is reported. For example, using well defined financial accounting rules the acquisition or upgrade of a major infrastructural component may be charged over its nominal lifetime. The distribution of charges over longer periods in this manner represents a consistent and fair assessment of situation since otherwise a large charge would appear to wipe out the profits of period in which it was incurred and thereafter seemingly play no further part in the economic fortunes

of the organisation. Thus to determine profit the structure of costs (i.e., capitalisation charges versus operational expenses) needs to be clarified.

Outgoings in projects (in particular those in the IT sector) can broadly be classified as capital or operational expenditures.[2] Although organisations have some freedom concerning how to classify costs there are rules [126] to which they must abide. Generally speaking, operational expenditure covers non-asset generating activity such as the assessment of whether or not a project should be initiated or costs incurred deploying and operating increments. Operational expenses are incurred within the current accounting period and are recorded on the income statement.[3] Capital expenditure, on the other hand, covers work that results in an asset (e.g., software package) or entities that must be acquired in order to support it (e.g., infrastructure), i.e., capitalised costs are incurred on assets that are not expensed in the period in which they were created but rather are depreciated or amortised over time. These are recorded in the balance sheet[4] and the depreciation costs are recorded over future income statements. Thus the central issue to determining which form of expenditure applies is to consider what the outcome of an activity is and what implications this has for ownership of the underlying asset. This helps resolve seemingly difficult issues that arise in practice. For example, acquisition of software or upgrades to existing infrastructure are capital charges whereas product support and maintenance (incl. bug fixing) are operational expenses. Equally consumption models may impact on this classification (e.g., SaaS is considered an operational expense since the asset is not on the balance sheet of the organisation using the service).

Most agile methodologies have difficulty in determining the point of capitalisation in their projects owing to the lack of clear boundaries that delimit capital and operational activities. In some respects DSDM provides a potential means of addressing this issue since it employs a phased model that enables such a separation of capital and operational expenses. This model, described in Chap. 4, segments the solution development process into Feasibility, Foundations, Evolutionary Development and Deployment phases sandwiched between Pre- and Post-Project phases. Most activities in the Evolutionary Development phase could in theory be considered capital charges whereas all other phases are mostly concerned with operational expenses as indicated by Fig. 3.1. Difficulty arises though in the explorative nature of Agile (e.g., prototyping) whose activities are generally expensed (e.g., prototyping and modelling that occurs during Evolutionary Development) suggesting perhaps that a percentage-based allocation should be used such as capitalising the Consolidation and Refinement subphases of a structured Timebox (see Chap. 4) as illustrated in Fig. 3.1.

---

[2] Capital and operational expenditures are often abbreviated to CAPEX and OPEX respectively.

[3] The income statement is an account of all revenues and expenses during a specific period of time (e.g., over the course of a year).

[4] The balance sheet lists all of the assets and liabilities of an organisation at a specific point of time. Assets are resources of net positive economic value that are owned or controlled by the organisation and liabilities are outstanding obligations that imply the transfer of funds or a commitment to provide goods or services.

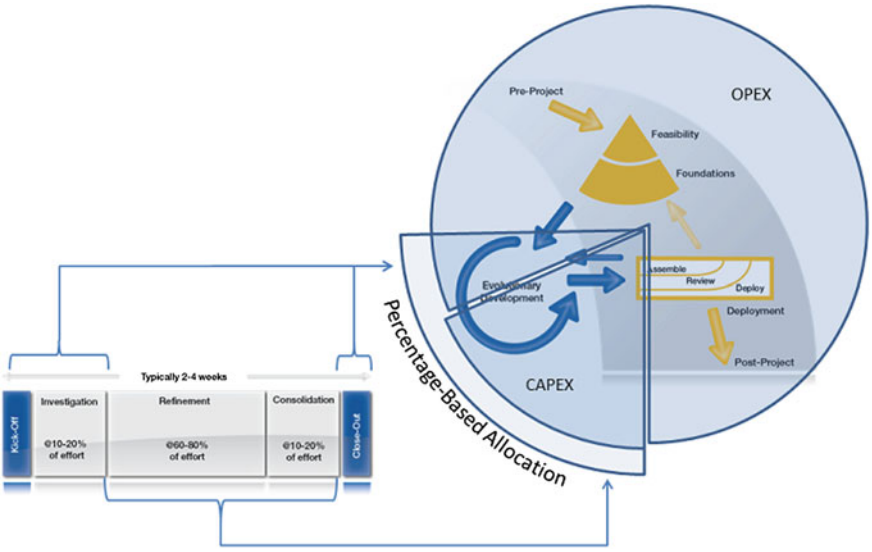

Fig. 3.1 Possible Partition of Capital (CAPEX) and Operational (OPEX) Expenditure for DSDM Projects. Adapted from [73]. Reprinted and modified with permission

The Scaled Agile Framework® argues that most asset evolving activity occurs on the Agile Release Train, a long lived group of agile teams working together to deliver programme level value, to which a number of dedicated support roles (and their related costs) are assigned [225]. Accordingly capitalisation at the Agile Release Train level may be expressed as a crude percentage of overall activity or be broken down to the epic and user story level once initial spiking has concluded that the work should be undertaken. Either story points or time may be used as the basis of allocation. This ensures separation of feasibility from implementation though other activities still remain that require specific treatment depending on the circumstances of the project (e.g., architectural epics and spikes). Applying similar principles to other methodologies (e.g., Scrum, XP) it is conceivable that a means can be found to account for costs though not without some burden on the team in terms of reporting.

In practice most product development organisation simply expense their costs (e.g., software development) whereas internal IT departments may use either form of classification as appropriate provided that this is done consistently. In the final analysis, taxation is often a consideration since expensing costs means that they are accounted for earlier and can be included in the profit calculation on which tax is based leaving some to suggest that capitalisation may appear to give an inflated impression of profits. It is therefore fair to say that profit is a matter of (accounting) opinion and thus care should be taken when using accounting based metrics to assess (agile) projects since these are strongly influenced by changes in the application of accounting principles (e.g., nature of amortisation or use of depreciation model).

3.4 Project Appraisal

Capital budgeting is concerning with the assessment of project viability against predefined criteria (with particular emphasis on the future value generation potential of a project) in order to determine where best to deploy a firm's limited resources and capital [63]. Since investment funds are capable of attracting interest over time, the timing of cash flows becomes relevant for investment decisions. Time value of money refers to the notion that present day funds are more valuable unit-for-unit than the same amount at a future point in time owing to their ability to accrue interest. Thus if an amount is capable of earning ten percent per annum then ninety cents today is equivalent to € 1 one year from now[5] since ninety cents today invested at ten percent per year amounts to approximately € 1. The premise of time value for money is that there exists an expected rate of return that remains constant for the period in question. In the corporate context this rate is usually linked to the weighted average cost of capital which broadly refers to the average of costs of financing an organisation (or departmental unit therein) based on its capital structure (i.e., the extent to which it is funded through a combination of equity and bonds). It follows that an organisation will therefore want to invest in projects that deliver a higher rate of return than that which represents its own financing or opportunity costs. Accordingly, the timing of cash flows of projects must be assessed in terms of a discount factor based on the weighted average cost of capital though in practice this rate is often adjusted as appropriate (e.g., to account for real rates of interest, inflation or to include a risk premium).

Capital budgeting involves the computation of future net cash flows based on expected revenues less costs (incl. working capital) all of which is discounted back to present day values. Thus projects will be approved if they are deemed to be capable of contributing a net positive value to the bottom line. When considering what constitutes a cash flow from a capital budgeting perspective there are a number of subtle details that must be taken into account (e.g., exclusion of sunk costs,[6] handling of marginal costs and the treatment of tax). Though important in practice these will not be considered in detail here owing to their technical and operational character and the fact that they are treated in entirely the same manner in agile projects as in traditional ones. The basis of net present value calculations is to divide the future into equal time periods (e.g., iterations) and to discount back net cash flows to the present day. Thus if a net amount of CF_i is realised in period i and the expected period rate of return is r then its contribution to the net present value is

$$NPV_i = \frac{CF_i}{(1+r)^i}$$

[5] This calculation is derived by $0.9 \approx \frac{1}{1.1}$.

[6] Sunk costs are those incurred before the decision-making process and which are not materially affected by the outcome of the decision.

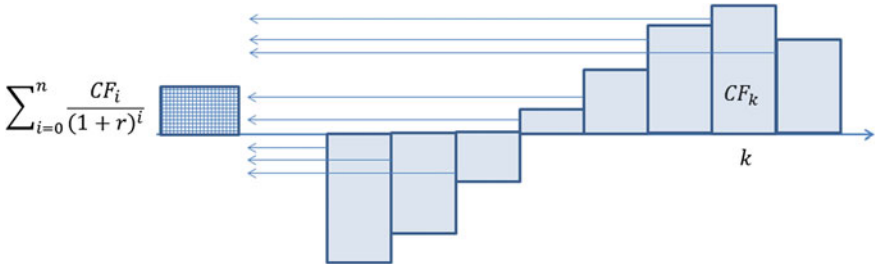

Fig. 3.2 Net Present Value of Project Cash Flows. Published with kind permission of © Alan Moran 2015

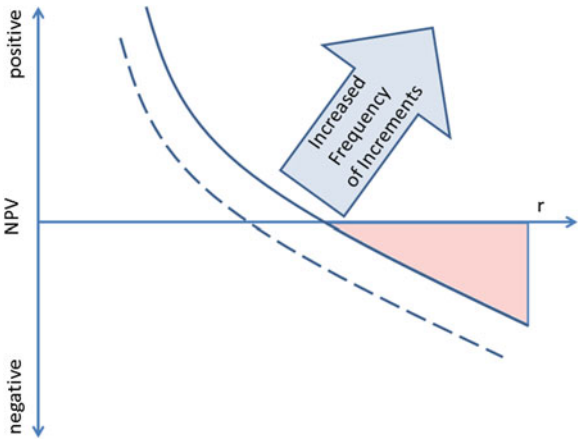

Fig. 3.3 Net Present Value Profile for an Agile Project. Published with kind permission of © Alan Moran 2015

or in other words an amount NPV_i today invested in a compounded fashion for i periods of time at rate r would accrue in value to CF_i which is just an algebraic reformulation of the above expression. Note that the expected rate of return is assumed to remain constant throughout and if this rate is annualised but the periods over which cash flows are measured are not, then the value of r needs to be appropriately adjusted.[7] The net present value for the project is the sum of all net present contributions of future cash flows as indicated in Fig. 3.2.

Figure 3.3 illustrates the NPV profile (i.e., a graph of project NPVs over a possible range of rates of return) for an agile project with an initial investment, project and post-project net cash flows over the finite lifetime of the solution but excluding decommissioning costs.[8] Analysis of the capital budgeting structure indicates that

[7] By way of example an annualised rate of return, r, equates to a compounded monthly rate of return of $1-(1-r)^{\frac{1}{12}}$.

[8] Decommissioning costs for some projects can be sufficiently significant to alter the shape of the NPV profile e.g., costs of taking a nuclear plant out of operation.

the NPV profile is influenced by the frequency of delivery of revenue generating increments and the distribution of benefits that each enables. This suggests that prioritising the most valuable increments first and releasing on a frequent basis will positively impact return on investment. Moreover, this also has the effect of pushing up the internal rate of return for an agile project which means that the decision to invest in such projects may continue be taken under increasingly more demanding expected rates of return thereby diminishing the basis for rejection of project proposals (see the shaded region of Fig. 3.3). By contrast a traditional project that delivers all its benefits on completion of the project (indicated by the dashed line in Fig. 3.3) fares less well and cannot be accepted at higher rates of return.

Another feature that emerges from agile project appraisal is the ability to continually assess return on investment with each increment delivered and to seek validation from the marketplace on the achievement of project objectives. Thus if a project does not appear to be meeting its targets then this feedback can provide validated grounds for cancelling a project rather than committing additional funds needlessly. This requires that there is adequate governance in place (e.g., the use of review points and benefits assessments exercises as outlined in Chap. 6) and that an incremental approach to determining return on investment is included (i.e., computation based on cumulative gains and costs of each increment using realised accounts once these figures have become available). Thus Agile in combination with appropriate governance and financial appraisal approaches provides a means of risk managing the delivery of value [101]. For example, there may come a point in the project where a decision must be made as to whether or not to continue (see the circle in Fig. 3.4) by which time certain benefits have already been realised. Thereafter the solution may remain in service (albeit without further development) and continue to accrue benefits, though it cannot be ruled out that market forces will not act in an adverse manner (e.g., a deterioration in the prospects of the target segment or entrants of

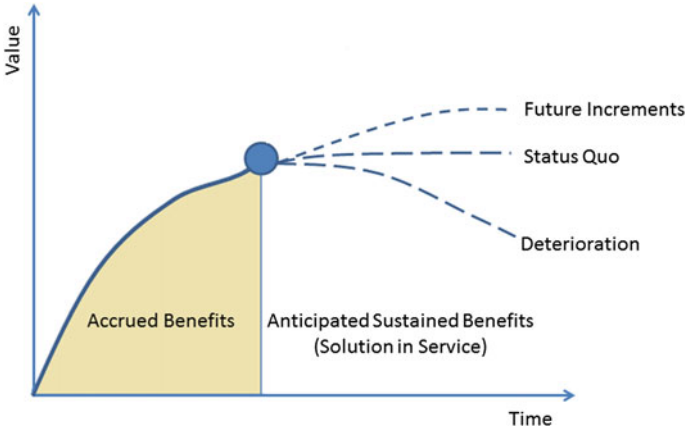

Fig. 3.4 Agile Value Curve. Published with kind permission of © Alan Moran 2015

new competitors with alternative solutions). On the other hand, a decision may be made to continue the project leading to a new increment that may offer additional new net benefits though owing to the prioritisation of value these are likely to be marginal when compared to those benefits already gained. Thus in the context of agile projects it makes sense to continually seek validation for ongoing efforts and to question whether or not there are opportunity costs involved by not engaging in other ventures. Either way, this is the point at which future investments must be reassessed against anticipated rates of return in order to determine if the pursuit of new value is still warranted.

3.5 Agile Contracting and Pricing Models

Inspired by the need to have certainty before committing to an undertaking many contractual arrangements fail to acknowledge the need to incorporate room for learning in projects. Consequently traditional approaches have focused on creating full up-front specifications replete with estimates on the basis of which costing or pricing is to be determined. This places considerable restriction not only on necessary changes that arise as a clearer understanding of project objectives and tasks is gained but also the changing circumstances of the external environment that may force a re-evaluation of the premises on which the project is based. Agile contracting, a topic about which most methodologies are surprisingly mute, must therefore address such matters in a manner that protects the flexibility of the underlying methodology but also points to obligations rather than pre-defined rigid outcomes. Such contracts may place more emphasis on the project outcome and vision, accommodation of learning, governance and risk and the nature of relationships and practices in the project. For example, there may be requirements concerning the formulation of acceptance criteria and obligations in terms of their verification and validation. Finally there should be a capping of total costs and an exit mechanism for situations where the contractual flexibility is unable to cope with conflict resolution.

There are many candidate pricing models available several of which are applicable to agile environments with only minor adaptations. For example, pricing based on time and materials is common and requires no alteration for agile projects whereas unit based pricing merely needs to be framed in appropriate terms (e.g., user story or epics). Alternatively more creative practices can be applied such as linking pricing to the delivery of value based on the end-usage of the increment. Where fixed priced models are applied it is common to add a risk buffer (e.g., ten percent of costs) to cater for unanticipated change which might be linked to team level metrics (e.g., velocity). There is, however, a prevailing view in the agile community that fixed-price contracting is incompatible with the agile approach though much of this criticism is linked to the manner in which scope becomes a fixed project parameter. Using the DSDM approach (as described in Chap. 4), it is, however, possible to fix quality, time and price but leave scope a variable. This admits sufficient flexibility with which to build trust between client and supplier especially given that progress can

be tracked at the incremental level with a sufficiently robust governance and risk management framework. Thus the resolution of the fixed-price dilemma requires more than simply adapting the existing contracting culture and involves ensuring that adequate structure and governance is in place in the agile environment.

3.6 Management Implication

Agile has implications well beyond the technocentric circles of solution development that impinge on how management is understood in terms of performance and financial accountability. It finds resonance with the Beyond Budgeting movement that challenges the traditional budget process on grounds comparable to those that seek to question traditional product development. Together both contribute significantly to the debate on how an organisation should align values, efforts and rewards.

From an accounting point of view the need to categorise costs (e.g., capital and operational expenditure) can impose significant burdens on agile teams that threaten to reduce their value generating potential. Some methodologies, however, admit the possibility of structural classification of costs that may be in line with wider accounting guidelines and practices (e.g., DSDM phased model and structured Timeboxes) thereby offering a non-intrusive means of accounting for costs. Such matters are important only insofar as profitability features in the assessment of Agile (e.g., use of profitability index as a project appraisal technique) though care needs to be taken to ensure these are not conflated with other stakeholder concerns (e.g., investors' desire not to overstate profit by using capitalisation charges or internal wishes to use operational expenses to reduce tax burdens). The use of net present value calculation of anticipated future net cash flows based on percentage enablement of benefits (without reference to accounting constructs) therefore remains a sounder means of appraising agile projects.

An analysis of NPV profiles of agile projects suggest that from a financial perspective prioritisation of value and delivery of frequent (revenue generating) increments afford higher rates of return, contribute positively to the bottom line and improve return on investment metrics. Furthermore, the validatory feedback enables sound decision-making in respect of continuation of projects assessed in the light of whether or not the anticipated new net benefits are in line with expected rates or return or if the opportunity costs of deploying teams elsewhere would suggest otherwise. Indeed this is a central principle of the DSDM approach (see Chap. 4) that advocates focus on business need together with on-time delivery. Finally, proactive governance and risk management measures ensure an appropriate balance of risk and reward throughout the project. This enables threats and opportunities to be identified and managed appropriately thereby avoiding wasteful deployment of resources and enabling exploitation of emergent opportunities. Avoiding locking funds in projects that delay the timing of cash inflows means that smaller amounts of funds can be distributed more effectively over multiple projects though the promise of a multiplier effect wherein projects achieve paybacks within the solution development period are

perhaps less convincing in all but exceptional cases. This, notwithstanding, the primary objective of maximisation of value and continual pursuance thereof can still be better achieved by the approach and frameworks offered by agile environments.

Most discussions concerning agile contracting remain framed by the traditional concept of a Waterfall style process wherein feasibility, specification and estimation provide the basis for contractual terms and conditions. This is a search for a false sense of security that is resolved by adapting existing contractual frameworks and pricing models to reflect the reality of uncertain outcomes, changing business environment and the necessity of cultures of trust and learning. Amending such arrangements requires an agile environment that supports iterative compliance measures and thus can only be achieved within an appropriate governance and risk management context. Ultimately, however, the success of project can seldom be found in a tightly worded contract but rather in the trust and integrity built up in the relations of the primary participants derived from a focus on business need, on-time delivery of quality solutions and continuous communication and collaboration.

Part II
Implementation

Amongst the myriad of agile methodologies, DSDM stands out as that approach most focused on being a vehicle of strategic implementation founded on agile programme and project management principles. It is therefore fitting to critically review DSDM in detail and consider how it contributes to a wide range of issues of concern to management including governance, quality and risk. Some attention is also afforded to configuration management as a holistic alternative to the widespread practice of assessing projects using performance metric bundles.

Chapter 4
Agile Project Management

Abstract This chapter considers the major elements of agile project management as perceived through the lens of the DSDM Agile Project Framework (which is related to DSDM Agile Project Management though there are significant differences). The introduction of this framework forms the basis of discussions elsewhere in this book concerning how wider matters including governance, quality and risk ought to be tackled in the agile context (though other disciplines covered in this book such as configuration management feature less prominently in the DSDM Agile Project Framework). Owing to the existence of an extensive body of literature concerning specific practices and guidance it is not necessary here to delve into operational details. However, where appropriate the standard DSDM literature is complemented with wider perspectives (e.g., agile success factors, role conflicts and compatibilities, implicit self-organisation roles) so that even the reader already familiar with DSDM may yet discover some new insights.

4.1 Introduction

As the first truly agile methodology and one of the only approaches with a focus on agile project management, DSDM [56, 70–73, 185] is perhaps best understood as a framework into which daily agile working practices can be structured and embedded. Its philosophy, principles and practices work together in concert to form an iterative and incremental basis for the creation and delivery of value. Whilst there is much that ought to be familiar to a traditional project manager it is its nuances and emergent driven practices that reveal its true agility.

The DSDM philosophy expounds the notion that business value emerges through communication and collaboration within an iterative and incremental approach based on firm foundations that delivers solutions to a well-defined level of quality and within a specific timeframe. This balance of control and autonomy is achieved through delegated decision-making within empowered teams operating to pre-defined governance guidelines. Understanding the transactional nature of power and embracing an adaptive planning approach enables such projects to cope much better with the

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_4

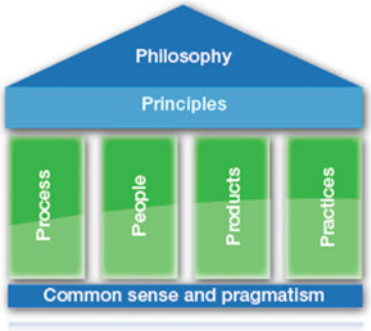

Fig. 4.1 DSDM Philosophy in terms of Principles, Process, People, Products and Practices © DSDM Consortium 2014. Reprinted with permission

inevitable change and risk that arise. Figure 4.1 illustrates the key features of the DSDM philosophy.

- *Principles*. Eight principles articulate the core elements of the philosophy in terms of focus on business need, on-time delivery, collaboration and communication, incremental and iterative work practices, a non-compromising stance on quality and the need to demonstrate control. These principles operate in concert and are reflected in the project activities (e.g., solution development, quality management).
- *Process*. A full lifecycle model that describes the transitions through phases that enable iterative and incremental practices to shape solution delivery.
- *People*. Facilitation of communication, support of collaboration and the composition and distribution of skills within project teams define the people parameter. These focus on creating synergies that are to be found in heterogeneous teams empowered to organise themselves.
- *Products*. A set of evolutionary and milestone artefacts cater for different project perspectives (e.g., solution development, managerial control, governance oversight) which when combined ensure that the solution evolves as intended.
- *Practices*. Structured techniques and activities (e.g., workshops, MoSCoW prioritisation, iterative development) that capture and express the underlying principles. These can be employed alongside the wider agile toolkit (e.g., refactoring, pair programming) as needs dictate.

The approach is conditioned by an ethos of common sense and pragmatism based on application of judgement and the taking into account of situational circumstances and their immediate practical consequences. In taking this stance, DSDM is clearly attempting to ensure that its processes do not undermine the very precedence of individuals and interactions over tools and processes that Agile promulgates. Although not unique to agile, many an attempt to formalize working practices has simply led to a blind adherence to theory over situational practice that has led to a culture of methodological tribalism.

DSDM turns the traditional project triangle (of time, cost and scope) on its head (see Fig. 4.2) by introducing fixed delivery timeframes (known as Timeboxes), fixing cost early on in the project (i.e., during the Foundations phase), setting quality

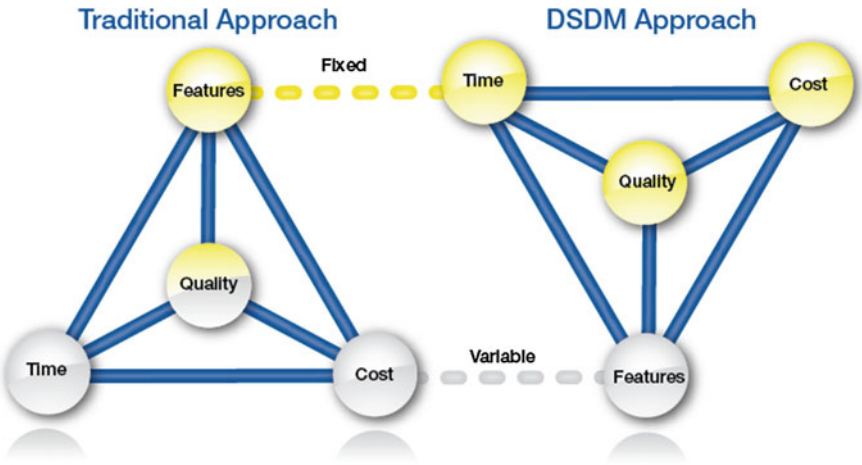

Fig. 4.2 Traditional versus DSDM Perceptions on Project Variables © DSDM Consortium 2014. Reprinted with permission

expectations from the outset and only permitting feature scope to vary. It must be said that the success (measured in project appraisal terms) of agile projects does owe much to this inherent flexibility. With the focus on feature scope, a forced ranking approach to requirements prioritisation[1] seems less appropriate thus necessitating a more flexible classification such as MoSCoW[2] to ensure that incremental delivery is addressed in a more discerning manner reflecting the most pressing needs of business. Indeed, the traditional belief that all project variables can be fixed is largely an illusion that fails to accommodate change or risk in a realistic manner. This is owing to the lack of active learning whereby estimation of effort is validated through experimentation and experience. Thus if estimation is found to be at fault then the entire premise of the "golden triangle" (of scope, time and cost) is undermined.

The DSDM principles are intended therefore to guide the team in its attitude towards working in agile projects. It is therefore to be expected that adherence to all principles is required in order for the full benefits of DSDM to be realised.

- *Focus on the business need*. Projects exist to service business needs in a timely fashion. Establishing a clear understanding of business goals and priorities and ensuring that continued support and commitment of all stakeholders is present constitute the key elements of this principle.
- *Deliver on time*. Since the agile project framework employs timeboxed practices, this principle implies that delivery should be accordingly structured and that timeboxes (and by implication deadlines) should not be extended. Thus with time (and quality and cost) fixed, scope becomes the only project variable.

[1] The classical example of forced ranking of requirements is the Scrum Product Backlog.

[2] MoSCoW is a common prioritisation acronym for Must, Should, Could and Wont.

- *Collaborate*. Fostering greater understanding through collaborative builds commitment and engagement. This principle advocates the breaking down of institutional barriers (e.g., co-location of integrated and heterogeneous teams where possible, empowerment through delegated authority practices and the establishment of a "one team" culture).
- *Never compromise quality*. Once expectations concerning quality have been established, they feature as a binding element of the project that require adherence and validation throughout. DSDM stresses the importance of quality not becoming a collateral casualty of other project variables (e.g., cost, time) though it does not suggest that the highest level of quality need always be attained.
- *Build incrementally from firm foundations*. The determination of an optimal solution often requires periodic revalidation of project assumptions to ensure ongoing viability of the proposed solution. Accordingly, the project lifecycle model provides for feasibility studies and other opportunities with which to lay a firm foundation for the project. Experience suggests that time spent in these phases is well invested.
- *Develop iteratively*. DSDM argues against specification stockpiling (e.g., full upfront designs) and appreciates experiential and learning effects that take place during the evolution of requirements whilst also acknowledging that change (e.g., new business priorities) often impinges on a project. Thus, the presence of feedback loops, the practice of adaptive planning and the culture of embracing change are all present in the process. Practices such as iterative development are particularly well suited for explorative development situations where understanding arises in an emergent manner.
- *Communicate continuously and clearly*. This principle endorses a range of communicative practices (e.g., stand-up meetings, facilitated workshops, modelling and visualization) that emphasize direct experience and human interactions over "colder" forms of communication (e.g., written specifications) and whilst these are not excluded their usage is curtailed to what is necessary and appropriate.
- *Demonstrate control*. Whilst preferring self-organisation and empowerment, DSDM nonetheless understands the value of project governance and the need to demonstrate control in a project. Multiple level adaptive planning and tracking of progress in terms of deliverables rather than ancillary project artefacts are typical practices in this context. Agile tracking and reporting (e.g., burndown charts, Team Boards) provide good examples of open, adaptive and people-centric artefacts that are found to be highly effective in practice.

Compared to other project management frameworks there is some similarity in terms of principles (and the practices derived from them). For example, the PRINCE2®[3] advocacy of continued business justification anticipates the need for revalidation of the business case throughout the project lifetime which together with its learning from experience (e.g., the transfer of lessons learned between projects)

[3] PRINCE2®, which stands for PRojects In Controlled Environments, is a traditional project management framework that adopts a phased though not necessarily incremental approach to product development.

mirrors some of the elements of the focus on business need, iterative development and communication principles. Other PRINCE2® principles (e.g., defined roles and responsibilities, tailoring of the project approach) are not explicitly cited as principles in DSDM though their practice is clearly evident elsewhere (e.g., role structures discussed later on). Similar remarks apply to the Project Management Institute® advice concerning commitment to project goals and practices [212]. Where both PRINCE2® and PMI® differ is in their principled adherence to phased delivery (e.g., management by stages and exceptions) and their overriding control structures that favour delegation of authority at the cost of team autonomy (e.g., requiring frequent approvals and limiting the scope of team level decision-making). Close inspection of such approaches thus reveals little of the empowerment or self-organisation culture that is known to enhance team commitment and motivation. Moreover, whilst a staged approach does not mean that a project must adopt a Waterfall stance, there is a tendency towards this direction in some traditional practices (e.g., the grouping of work in technical stages as defined by specific skill sets [196, Chap. 10]).

This is of course not to say that traditional frameworks cannot be successfully combined with agile either as part of a transition strategy towards more Agile ways of working or as a practice in itself [260]. Indeed the DSDM Agile Programme Management Framework makes explicit accommodation of both agile and non-agile practices. However, there are challenges to be found in the movement away from top-down direction, the fostering of a genuine culture of trust and respect and relinquishing of the need for written tracking and status reports in favour of more open and effective lightweight forms of communication. In this respect, the DSDM Agile Project Framework appears to strike a healthy balance between necessary control and governance of projects whilst at the same time enabling sufficient empowerment for team members to engage effectively.

There has been much investigation into what constitutes success factors in agile projects. This has revealed that a corporate culture of mutual trust and cooperation is perhaps the most dominant success factor during the adoption phase of agile processes (i.e., when organisations are transitioning to agile) though its continual presence thereafter also helps. Another major factor is training and education founded on a culture of learning. Indeed, there is mounting evidence that the explorative and learning nature of agile processes is acutely suited to innovative environments and is believed to contribute significantly to the competitive advantage of an organisation (see Chap. 2). Moreover, organisations that successfully capture and share the tacit knowledge of their employees (e.g., collaborative practices such as pair development, internalization of experiences in the form of retrospectives) appear to particularly excel with agile approaches. Within the specific context of DSDM a number of other factors have been proposed:

- *Embracing the DSDM Approach*. As might be expected, an understanding of and a commitment to the DSDM approach is an inherent success factor in its application.

This is particular true for organisations transitioning from existing frameworks whose language and approach bear some resemblance to the DSDM methodology.[4]

- *Effective Solution Development Team*. This is a composite factor that comprises of the following elements:
  - *Empowerment*. This encompasses the extent to which decision-making is delegated which is often set out during the Foundations phase (e.g., ability of the Solution Development Team to make changes to the implementation but not the scope of a requirement). A key criterion in the assessment of this factor is timeliness of delivery and the absence of the need for continual approvals to higher authorities. Indeed systems thinking (see Chap. 1) suggests that such communication channels introduce delays into decision-making processes that may impair the quality of decisions made. Thus the central paradigm of self-organisation (see Chap. 10) features strongly in this success factor.
  - *Stability*. Agile thrives on implicit knowledge exchanged through direct experience [256] and rich communication channels [50, Chap. 3] both of which are collectively weakened in the presence of team churn.
  - *Skills*. As a collective the Solution Development Team must possess the necessary skills to be able to deliver the solution. The ideal of a T-shaped skilled individual (as described later in this chapter) who possesses good communication skills applies here though the wider set of social skills beyond mere communication also plays a crucial role (as described in Chap. 11).
  - *Size*. In keeping with most agile methodologies the optimal Solution Development Team lies between five and nine individuals though this figure excludes external advisors (e.g., Advisors and Coaches as well as the project level roles that shape and co-ordinate the project). It is generally believed that communication overhead is directly proportional and individual effectiveness inversely proportional to team size [50, Chap. 4].

- *Business Engagement*. The separation of business and technical solution activities and personnel appears to arise in non-agile environments where a staged approach is adopted (e.g., business defines requirements followed by a stage of their technical implementation) though this is an organisational rather than a methodological issue. Indeed, the absence of business during solution development creates substantial risk in terms of understanding of requirements and timely delivery of the correct solution:
  - *Commitment of Business Time*. It is considered essential that business allocate time for the project whether this be in the early phases where the vision and high-level requirements are set out or later on when it comes to detailed analysis and validation. Ultimately this is a matter for project governance through which business can demonstrate its commitment to the evolving solution and provide assurance of its continued validity.

[4] At times DSDM is phlegmatic about the manner in which the methodology is complied with, suggesting that whilst the entire methodology can be used some of the time, some of it can be used all of the time.

  - *Active Involvement of Business Roles*. Involvement of business roles often implies their physical presence within the Solution Development Team though this brings a mixture of advantages and disadvantages.[5] This notwithstanding, the availability for clarification of business roles at all times is a major benefit to decision-making and productivity
  - *Supportive Commercial Relationship*. This factor applies particularly to relationships that are subject to contractual conditions (e.g., the integration of external suppliers) where care must be taken not to burden the process with arduous terms of agreement (e.g., the injection of delegated authority approval processes).

- *Iterative Development, Integrated Testing and Incremental Delivery*. This factor best expresses the value of feedback loops that support the evolving solution and its assessment of fitness for purpose to ensure a manageable transition of the solution into usage. As a risk mitigation measure its capacity to adapt to changing circumstances and apply corrections where deviations from expected outcomes are encountered is also widely acknowledged in agile circles.
- *Transparency*. The clearest sign of progress is a (partially) working solution and it is this measure of delivery that agile methodologies, including DSDM, prefer. Supporting metrics may also be employed where these make sense and this can be further enhanced by project configuration management, a discipline described in some of the DSDM literature though no longer formally part of the DSDM Agile Project Framework. Common metrics in IT projects include test coverage of code expressed as a percentage of total code or key indicators of quality based on static code analysis. Despite their technical intricacies, reasonable targets can still be set (e.g., on the basis that more testing reduces the likelihood of undetected defects, one might aim for 50 % test code coverage). Agile is replete with lightweight and meaningful transparency tools that can also be deployed in almost any project setting (e.g., the most common transparency tools include Kanban boards and burndown charts).

4.2 Core Practices

Whilst employing a wide range of techniques found in many other agile methodologies, DSDM places particular emphasis on a core set of techniques which are outlined in this section. These include Workshops, MoSCoW Priorisation, Iterative Development, Modelling and Timeboxing.

[5] The value of osmotic communication, e.g., the benefit of overhearing relevant conversations, must be balanced against the disadvantage of drafting, i.e., hearing of unwanted and perhaps noisy information that colocation brings with it.

4.2.1 Workshops

A workshop[6] is a meeting with clearly defined objectives wherein participants, gathered together for a common purpose and who may be supported by an independent facilitator, engage in discussion in order to solve problems, generate ideas or determine courses of action. Facilitation requires distinct roles (as described in Appendix C) for workshop ownership, organisation and facilitation and where necessary maintaining a record of decisions and unresolved issues. The facilitator should never be the topic owner and the topic owner should not be permitted to frame to steer the discussion towards a predefined outcome. Workshops usually require sufficient numbers of participants to be effective in terms of group dynamics (a commonly cited minimum is four) but can be very large complex events involving over a hundred people and using several co-facilitators. Participation in the elaboration of ideas and solutions engenders commitment and motivation towards their realisation and can help mitigate misconceptions and misunderstandings that hamper the solution discovery process. The presence and immediacy of appropriate participants along with the verbal nature of discussion contributes to highly effective communication assisting the building of consensus and the clarification of issues. Finally, workshop events can build team spirit and even energize a project as a collective actionable consensus emerges.

There is no single way of being an effective facilitator though the popular taxonomy of hierarchical (i.e., directing the process and leading from the front), co-operative (i.e., sharing power, enabling and guiding) and autonomous (i.e., creating the conditions to support the group) is commonly used to classify approaches. Although there exists circumstances in which each may be considered to be the preferred mode, clearly the co-operative and autonomous modes are better suited for agile environments [104]. Irrespective of approach a facilitator must be capable of organising and planning the event, helping the group establish a common meaning and understanding, challenging held views and behaviours, dealing appropriate with emotional responses and creating a climate of trust, respect and integrity. At a personal level facilitators must project confidence with an assertive but respectful attitude and may employ deep listening techniques[7] in order to be in a position to correctly capture the views of the group. Furthermore, a facilitator must be capable of observing and responding appropriately to individual behaviours and signals (e.g., interpreting body language) whilst permitting the group themselves to find the solution to their problem (e.g., allowing time for discussion). Not infrequently conflict will arise in facilitated workshops which may either be attributed to the topic at hand or the personalities involved in the discussion. Conflicts may therefore be categorised as cognitive (i.e., intellectual points of discussion concerning the topic at hand) or

[6] Whereas the DSDM Agile Project Framework refers to workshops, earlier versions of DSDM (e.g., Agile Project Management) referred specifically to *facilitated* workshops as a core practice.

[7] Deep listening is an interview technique wherein the questioner engages fully in listening to responses without attempting to judge them or control the direction of the conversation. This requires not only letting go but also an air of calmness and open reception towards the other party.

affective (i.e., tensions arise from conflicting personality traits). Less important than the conflict itself is it origins (e.g., cultural, political, different backgrounds, experience or perspective) and it falls to the facilitator and participants to constructively work through such matters or deal with them outside of the workshop if necessary.

4.2.2 MoSCoW Prioritisation

MoSCoW prioritisation chiefly applies to all requirements and tasks but can also be used for acceptance criteria, tests and other related artefacts and activities each of which can be classified in terms of Must (e.g., inability to deliver would constitute failure, may be considered unsafe, illegal or not viable), Should (e.g., important though not vital, there exists a workaround if not present), Could (e.g., desirable though less important, lesser impact on omission) and Won't (e.g., considered out of scope for the timeframe under consideration). Stipulating litmus test rules for MoSCoW categories is considered a best practice. For example, were an item not to be delivered the response of which would be to cancel the project, then the classification as Must is to be deemed appropriate. Escalation paths and decision-making capacity should also be agreed early on in the project. The set of all Must items constitutes the Minimum Usable Subset and ought to make up approximately sixty percent of the timeframe effort. Note that it is a common mistake, symptomatic of lack of requirements decomposition, that most if not everything is classified as Must. The Should items usually contribute a further twenty percent of the effort leaving the remaining Could items to act as contingency in planning[8] (see Fig. 4.3). MoSCoW prioritisation applies at project, increment and Timebox levels and where appropriate care should be taken to take sensible account of the impact of dependencies (e.g., a Should item that depends on a Must item requires priority escalation) and testing (e.g., where prioritised testing is employed).

4.2.3 Iterative Development

Iterative Development concerns the emergence of a solution of known business value wherein each cycle brings the realisation of that value progressively closer. DSDM encourages the cycle of Thought, Action and Conversation as described in [73]. The term cycle refers to the traversal of solution development activities (i.e., there may be several cycles within a Timebox) wherein a conversation is initiated on the basis of what needs to be done and how this will be achieved and reviewed. Conclusion of a cycle may result in a solution of acceptable quality or may trigger a new cycle to continue working on the emergent solution. The manner in which conversations take

[8] In earlier versions of DSDM there was an explicit identification of Must and Should items with the Business Case though this link is rather more implicit in the DSDM Agile Project Framework.

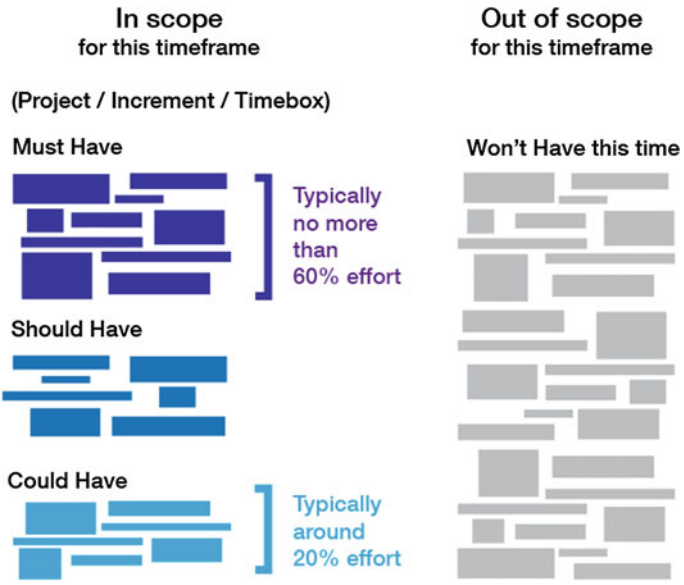

Fig. 4.3 Breakdown of MoSCoW Categorised Items © DSDM Consortium 2014. Reprinted with permission

place may vary choosing to focus on requirements (e.g., conducting conversations around functional, non-functional or usability aspects) or on solution architecture in terms of delivery on a horizontal, vertical or combined basis. Whilst horizontality refers to the delivery of the solution by layers resulting in a visible solution only when the final layer is complete, verticality refers to the delivery of solution slices through multiple layers resulting in partial solutions. Combined, on the other hand, refers to any appropriate combination of horizontality and verticality such as alternate delivery of thin horizontal slices to ascertain breadth together with vertical slices to delivery functional features.

4.2.4 Modelling

Modelling refers to the representation, simplification, abstraction and consistency checking of concepts through the use of models, prototypes or mock-ups.[9] As a learning process it solicits clarification and builds consensus around emergent solutions through the articulation of intended outcomes (e.g., architectural or process diagrammes, storyboarding, scaled models). In this context it is worth bearing in mind that the common interpretation of an architectural spike is that of a model

[9] These remarks concerning models apply equally to the capture of current and future states of the system.

intended to gain deeper insights but which will not be used as part of the final solution i.e., it is a throw-away learning construct. Modelling occurs in the context of its intended audience and must therefore take account of the relevancy of information (in the solution domain), its functionality (e.g., in terms of features and processes), locality (e.g., context of business operations), significance of events (incl. timeliness) and the business objectives and strategy as they relate to the project. In terms of the process phases, it is likely that Feasibility requires some degree of prototyping to better understand the implications of proposed designs and that these are developed into high-level process and architectural descriptions during the Foundations phase. Thereafter most modelling occurs at a tactical level during Evolutionary Development with minimal modelling during Deployment (e.g., infrastructural or topological models).

4.2.5 Timeboxing

A Timebox is the name given to an iteration in DSDM. Each increment contains one or more Timeboxes each of which is defined as a fixed time period with well defined objectives the result of which is assessed in terms of outcomes rather than completion of task based activities. Generally speaking a Timebox is a two to four weeks duration exercise peppered with review points in the case of structured Timeboxes (i.e., one review on completion of Investigation, Refinement and Consolidation as described below) during which solution quality and quality assurance is conducted (see Chap. 7 for a discussion of these terms). There are two forms of Timebox, the structured timebox, depicted in Fig. 4.4, and the free format Timebox shown in Fig. 4.5.

In DSDM the structured Timebox is broken down into parts, each of which involves specific functional activities and is concluded with quality or review related measures:

Fig. 4.4 DSDM Structured Timebox © DSDM Consortium 2014. Reprinted with permission

Fig. 4.5 DSDM Free Format Timebox © DSDM Consortium 2014. Reprinted with permission

- *Kick-off*. This opening session is concerned with establishing an understanding and acceptance of the objectives. In particular, elements of the delivery plan are assessed in terms of feasibility, availability and commitment of team members and dependencies.
- *Investigation*. This phase explores and clarifies the requirements in greater detail, determines appropriate acceptance criteria for the deliverables along with success criteria for the Timebox itself. Understanding of the solution may be deepened through the creation of prototypes. On completion of this phase a detailed understanding of requirements, their dependencies, priorities and risks together with a validation of the Timebox plan and assessment of resourcing should have emerged. Any review details are recorded in a Timebox Review Record.
- *Refinement*. The majority of the solution development and testing that occurs during this phase is informed by the findings of the investigation phase. The review point that concludes this phase identifies any necessary finishing work and determines a point beyond which no new work should be started. Like the investigation phase, review details are noted in the Timebox Review Record.
- *Consolidation*. Ensures that any items pending completion are attended to and that acceptance criteria have been satisfied. Final testing and quality reviews are conducted and incomplete work is removed.
- *Close-out*. Formal demonstration and acceptance testing of all deliverables that is often followed by a retrospective to determine if there are any lessons learned that can be applied to the next Timebox. The formal sign-off of the Timebox deliverables occurs in this phase and a decision is taken regarding incomplete work (e.g., postponement to future Timeboxes or descoping).

An alternative to the structured Timebox is to use a free format Timebox (see Fig. 4.5) that contains only Iterative Development sandwiched between Kick-off and Close-Out phases.[10] The advice provided in the DSDM Agile Project Framework literature to retain any number of formal or informal review points during the Iterative Development phase is clearly intended to avoid weakening project governance. Since in reality, however, it seems more plausible that such a Timebox format

[10] The free format Timebox is the direct mapping of a Scrum Sprint.

would be chosen precisely to reduce the overhead of a structured Timebox, it is questionable if this advice would be heeded in practice commendable though it is. Another rationale for the existence of the relatively new free format Timebox[11] is, of course, to enable the integration of DSDM with other frameworks that do not forsee its more structured sibling (e.g., Scrum). This notwithstanding, either form of Timebox structure imbued with whatever level of review is appropriate to the circumstances is entirely valid as there is no stated preference in DSDM towards one form of timeboxing over the other.

4.3 Lifecycle

The DSDM process is arguably unique amongst agile methodologies in the manner in which it frames product development in a project management setting. Its ability to accommodate a wide range of other frameworks and practices together with the attention it gives to both project management (e.g., governance, planning and control, risk management) and product development concerns (e.g., quality, testing) suggests that the process has a wide range of applicability outside of the traditional IT domain. The process, depicted in Fig. 4.6, comprises of the following phases and indicates permissible transitions between them:

- *Pre-Project*. This phase is primarily concerned with understanding the project objectives and business drivers and ensuring that the project has been initiated in the correct manner. In the portfolio or programme context, this phase may also contribute to an early assessment of the potential priority of the project.
- *Feasibility*. During this phase both the technical feasibility and business cost-effectiveness are assessed with a view to determining if the project should be undertaken. In small projects the Feasibility and Foundations phases are often merged into a single phase as suggested by Fig. 4.6.
- *Foundations*. The initial work of the Feasibility phase is developed further in terms of an assessment of the business, solution and management foundations during the course of which a business case, high-level requirements, proposed solution architecture and development approach, a high-level delivery plan and management approach are all elaborated. This essentially forms a baseline for the project that may be revisited later in order to re-assess or revalidate underlying assumptions.
- *Evolutionary Development*.[12] This phase constitutes the main focus of iterative development that sees a technically appropriate solution evolve towards the needs of business.

[11] Earlier versions of DSDM made reference only to variations of the structured Timebox.

[12] Earlier incarnations of the DSDM process separated the Evolutionary Development into Exploration and Engineering phases.

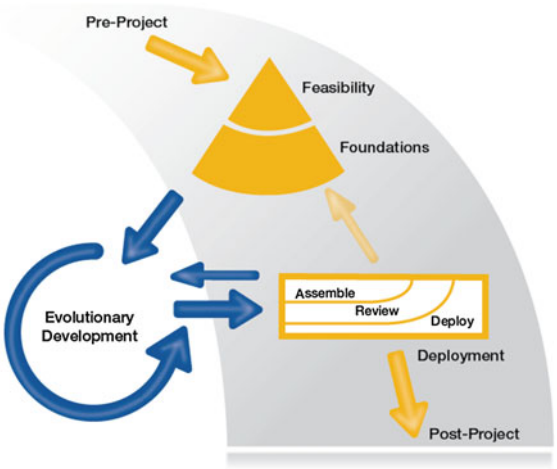

Fig. 4.6 The DSDM Process © DSDM Consortium 2014. Reprinted with permission

- *Deployment*. Periodically during the lifetime of the project, the underlying solution will reach a state where it may be transferred into operational use thereby fulfilling part or all of the business needs. This phase involves the following subphases:
  - *Assemble*. This requires the consolidation of all artefacts deemed relevant for the deployment of the solution. For example, the deployment of a business process may require a new software package, documentation for users and a communication to all stakeholders.
  - *Review*. This is essentially a quality gate that requires approval of the correctness and completeness of the assembled assets and offers an appropriate point for reflection on the project increment. Most organisations institutionalise such measures as checklists or more formal approval boards.
  - *Deploy*. This phase concerns the actual act of transition of assets into operational use. For example this might entail installation and configuration of a software package, training of users and release of a communication.
- *Post-Project*. This phase commences after the final increment has been deployed and is chiefly concerned with an assessment over time of the benefits accrued. Assessment of earlier increments may already by this time have taken place in which case they may be consolidated into a single report or reported on separately as needs dictate.

Understood in conceptual terms, this process sets out a structure rather than explicit working practices. For example, Assembly during the Deployment Phase neither precludes nor recommends that components of a solution be integrated for the first time. Indeed, to do so would be to delay learning and increase risk. It may therefore be assumed that the DSDM process is consistent with common practices found in agile environments (e.g., continuous integration, delivery and deployment).

4.4 Roles and Responsibilities

The character of project roles in Agile varies according to methodology with each expressing varying degrees of granularity in respect of specialism versus generalism. Generally speaking, it is deemed acceptable to have a mixture of specialists and generalists within a team provided that the team as a whole possesses the necessary skills in order to the deliver the solution. Whilst most methodologies are less inclined to define specific roles, preferring instead to appeal to broad functions within the team (e.g., servant leader, supporting coach, business representative), the DSDM Agile Project Framework proposes over ten roles emphasizing the need for empowerment within the team, respect for domain specific knowledge and skills as well as personal responsibility and courage when working together. This has the effect of making role related risks clearer early in the project. For example, if it transpires that a Business Ambassador role cannot be filled then the validity of decisions made at the Timebox level poses a risk that threatens to undermine the fitness for purpose and use of the final solution. Simply knowing that the role is not occupied suffices to identify the risk before work has even commenced. These roles are divided into areas of interest (i.e., business, solution development, managerial and process support) as listed in Table 4.1 and depicted in Fig. 4.7. Detailed descriptions of all roles can be found in the Appendix C. The distinct levels of roles reflect their engagement in project activities. For example, project level roles assume responsibility for the strategic and high-level activities and do not engage in the detailed level planning and execution which is delegated to the Solution Development Team.

As is usually the case with roles, multiple roles can be assigned to an individual (e.g., in small teams, a senior solution developer might also assume responsibility for team leadership). If extended to project management, then the assumption of project management functions by a solution developer alongside their normal technical activities is referred to as the "conformist paradigm" though this is rarely an exemplary practice [188]. Equally a role might also be divided up amongst several individuals (e.g., solution developer roles are commonly assumed by several distinct

Table 4.1 Project Role Matrix

	Managerial	Interests	
		Business	Solution/Technical
Project Level	Project Manager	Business Sponsor Business Visionary Business Analyst	Technical Co-ordinator
Solution Development Team	Team Leader	Business Ambassador Business Analyst	Solution Developer Solution Tester Business Analyst
Supporting		Business Advisor	Technical Advisor

Fig. 4.7 Agile Project Framework Roles © DSDM Consortium 2014. Reprinted with permission

individuals each focusing to varying degrees on their own specialism whilst still participating in the wider sense).

In addition there are a number of ancillary roles (not listed in Table 4.1) that typically play a supporting role in relation to one or more of the above categories. These include the Workshop Facilitator (who themselves may be supported by a technical scribe) and the DSDM Coach though other minor roles are also cited in the DSDM literature (e.g., Workshop Owner, Technical Scribe and Observer all of which are described in [73, Chap. 9] with earlier versions of DSDM also including a specialist Configuration Manager though this role could have been assumed by the Technical Co-ordinator). Some roles traditionally found in solution development environments are absent from DSDM as indeed they are from other agile methodologies. These include the Quality Assurance Manager (subsumed under the Technical Co-ordinator) and the Test Manager both of which are deemed largely unnecessary on account of the integrated nature of quality and testing in agile environments.

The project level roles, listed in Table C.1 of the Appendix C (the description of Business Analyst is deferred to Table C.2 of the Appendix C), are collectively accountable for project direction, coordination and governance. In particular the championship of the business case, its resourcing and funding and the quality assurance of the delivered solution lie within the remit of these roles. Some of these roles relate to those found on project steering committees in most organisations. In contrast to traditional project environments, however, the project level roles should support

and facilitate the Solution Development Team so that they are empowered themselves to deliver the solution rather than adopt a command-and-control stance (e.g., imposition of strict management by exception style control structures, top-down corrective planning on behalf of the Solution Development Team). The servant leadership character of project level roles is intended to foster self-organisation and empowerment which together with openness, transparency, trust and communication ensures that collective progress is traceable.

Solution development roles (see Table C.2 of the Appendix C) have an operational focus on the detailed planning, developing, testing and deploying of the solution. With the exception of the advisors, who are called upon when needed, these roles are committed throughout the project. Reflecting one of the key success factors of DSDM teams discussed earlier, stability of the core team is essential and constitutes both a risk and a matter for project governance if this is not the case. As a collective, the team must possess sufficient skills to deliver the solution. At an individual level, the DSDM Agile Project Framework promotes the notion that individuals with "T-shaped skills" (i.e., a deep understanding of their own discipline and a wider appreciate of how it interacts with others disciplines as depicted in Fig. 4.8) are most conducive to collaboration.

Supporting roles (see Table C.7 of the Appendix C) are generally assumed by individuals who are on the periphery of solution development (e.g., Business or Technical Advisors) and who may not even be members of the organisation (e.g., external DSDM Coaches). These ancillary roles are called into action only when needed and act to support (rather than to direct) decision-making. The DSDM Agile Project Framework provides no specific guidance about combination of roles beyond a few points of advice (e.g., Business Visionary ought to be an individual rather than a role dispersed over several people). However, by appealing to general project

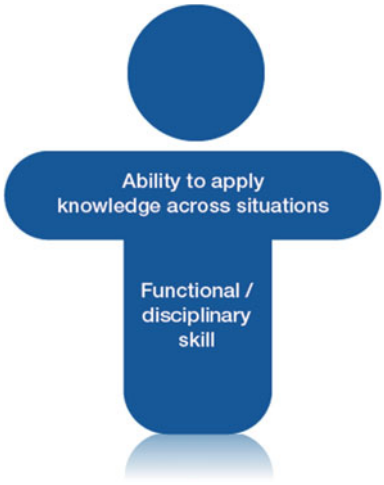

Fig. 4.8 T-Shaped Skills

Fig. 4.9 Possible Role Combinations. Adapted from [73]. © DSDM Consortium 2014. Reprinted and modified with permission

governance principles such as segregation of duties, it is possible to make a number of observations, suggestions and recommendations each of which needs to be assessed in the context of the project (e.g., size of organisation, complexity of undertaking). Note that all remarks pertaining to possible combinations of roles or those that may give rise to conflicts are indicative of the sort of thinking that the fine granularity of the DSDM roles enables.[13] Thus DSDM itself is not prescriptive in this respect but rather expects that sound judgement, linked to the specific circumstances of a project, be applied at all times. Consider first those roles which may be reasonably combined unless circumstances dictate otherwise (see Fig. 4.9):

- *Business Sponsor and Business Visionary*. It is not unreasonable to combine these roles since they are primarily concerned with the strategic perspective. This combination is less suited in complex projects where more than one Business Sponsor might be involved since the Business Visionary role may introduce conflicts of interest or bias. In ideal circumstances, however, the Business Visionary should be a single individual free of any role conflicts that might interfere with their participation in the project.
- *Business Visionary and Business Ambassador*. In some cases, both of these roles could be assumed by the same individual since they both concern the business

[13] Much of what follows from here to the end of this section is not part of the official canon of DSDM.

case, requirements and liaison with stakeholders. There is a potential for tension between strategic and tactical considerations and this is particularly so when the project advocates a change away from existing practices where existing interests are threatened.

- *Business Analyst and Solution Tester*. The joining of these roles ensures that validation and verification are performed by the same individual who also has a detailed knowledge of the relationship between business and technical domains. Any merging of roles with responsibility for quality and testing should, however, be avoided as these functions are usually kept distinct. For example, a Business Analyst who is also a Business Visionary or a Solution Tester who is also a Technical Co-ordinator might invalidate this combination of Business Analyst and Solution Tester.
- *Team Leader and Solution Developer*. This is a not uncommon form of "first amongst equals" leadership wherein a single solution developer assumes the lead developer function for part or all of a project. This approach is particularly well suited if that individual has been freely elected by the team to assume that leadership role or has earned it by virtue of experience or expertise. Moreover the precise individual who bears the Team Leader role may be permitted to change during the project reflecting the different focus that project phases may have during its lifetime.

As indicated above, combinations should not in general be considered transitive i.e., the ability to combine two distinct but overlapping pairs of roles does not imply that all three roles should be combined. For example, The combination of Business Sponsor and Business Visionary and the combination of Business Visionary and Business Ambassador may each have their own perfectly acceptable rationale. However, the combination of all three roles might justifiably give rise to concern that too much power is being consolidated in a single individual which in turn gives rise to a project governance issue. More problematic are the combinations of roles where the conduct of activities requires that they be performed by different individuals. The following are some examples where issues might arise (see Fig. 4.10):

- *Business Visionary and Technical Co-ordinator*. This conflicts with the principle of segregation duties wherein one party must determine what is to be accomplished, whilst another determines how this is to be achieved. Such a combination may weaken the ability to critically assess the feasibility of a solution or risk injecting technocratic considerations into the business case.
- *Business Sponsor and Project Manager*. These roles entail very different activities and their combination endangers a common project governance principle wherein a project manager must report to a steering committee whose independence is therefore questionable.
- *Team Leader and Project Manager*. This combination needs to be handled with care since it impacts the different levels of planning that these roles are expected to assume and might bring into conflict different styles of leadership. Thus the issue here is not the combination of roles *per se* but rather its implications for the agile approach and the risk of regression into non-agile project approaches. This

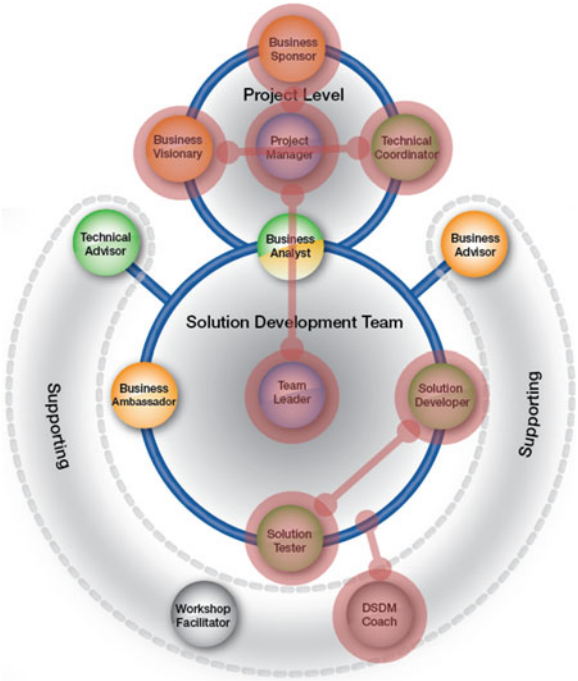

Fig. 4.10 Potentially Conflicting Role Combinations. Adapted from [73]. © DSDM Consortium 2014. Reprinted and modified with permission

notwithstanding, it is not uncommon in small projects to see a Solution Developer (who may also be a Team Leader) assume Project Manager responsibilities on the side. This combination should not, however, become a license for traditionalists who are still coming to terms with agile to centralise command-and-control structures around them at the cost of self-organisation and team empowerment.

- *Solution Developer and Solution Tester*. The joining of these roles violates the central principle of segregation of duties that enables an individual to independently assess the suitability of a solution as well as a violation of the independent testing principle as described in Chap. 7. This does not, however, preclude a Solution Developer testing their own output (e.g., unit testing, component testing) or assuming testing responsibility for the output of another Solution Developer. Rather it is suggested that individual Solution Developers should not have the final word on testing their own work.
- *Coach inside the team*. An internal coach can lack impartially and could be perceived rightly or wrongly by some team members as being partisan. Accordingly, it is helpful to locate Coaches from elsewhere in the organisation or outside of it. Such individuals, provided that they bring with them deep understanding of the methodology, its application and adaptation, can also transfer best practices from other environments into the project.

These comments should be understood as guidelines and clearly there are circumstances where it is simply not possible to comply with all of them. Equally there are situations in which such combinations might be perfectly fine provided that there is an awareness of the issues involved (e.g., coaching may emerge from within the team in the form of impliciting mentoring as described later in the discussion on implicit roles). Accordingly judgement is required to apply them appropriately. Equipped with an awareness of the potential issues involved in combining roles it becomes possible to identify and manage the risks that arise from the separation and combination of roles.

Within a project there are cross functional domains that do not fall entirely into the remit of an individual role but must be borne by several distinct roles. Although these domains are covered in detail elsewhere, it is helpful to sketch out which roles are primarily involved:

- *Testing*. Responsibility for testing lies with the Business Ambassador who must determine appropriate acceptance criteria. The primary roles concerned with operational testing are the Business Analyst and Solution Tester who cater for validation and verification respectively (see Chap. 7). Between these two roles the solution is assessed for fitness for use and fitness for purpose and together these determine the ultimate suitability of the solution. Needless to say, several other roles also contribute to testing in a secondary capacity to their primary function. For example, the Technical Co-ordinator (and perhaps also the Team Leader) should ensure that testing standards and practices are engaged in, the Solution Developer will likely perform testing at a unit or component level and the Business and Technical Advisors may provide specialist input to specific testing scenarios and cases.
- *Quality*. As discussed in detail in Chap. 7 the quality attributes that find strongest expression in the DSDM Agile Project Framework concern the features, performance and maintainability of a solution. These lie chiefly in the domain of the Business Visionary (and to an extent the Business Ambassador) together with the Technical Co-ordinator who together bear much of the responsibility for quality. Clearly other roles are also involved including the Business Analyst, Solution Developer and Business and Technical Advisors who must furnish solutions consistent with the quality expectations set out early on in the project.
- *Risk*. Agile risk management (see Chap. 8) advocates both that risk management is the responsibility of all team members and that one role should be accountable for its practice. In the DSDM Agile Project Framework the Project Manager best fulfils the role of risk manager who must ensure that risks are collectively discussed and addressed. Additional roles that are important in risk management include the Business Visionary who must keep abreast of business risks in the wider project environment and the Technical Co-ordinator who has oversight for technical risks in the solution technical architecture. Finally the Business Analyst, Solution Developer, Solution Tester and Business and Technical Advisors can each assist in the identification of risks in their respective domains.

Though not discussed in the DSDM literature, the nature of self-organisation in agile teams has a profound and subtle impact on how roles are assumed and

Table 4.2 Tacit Roles in Support of Self-Organisation. Adapted from [113]

Role	Description
Mentor	Provides guidance and support (esp. during the initial phases of agile adoption) to ensure that agile practices are understood and correctly applied. Encourages the development of self-organisation within the team
Co-ordinator	Represents the team in relations with the customer by managing expectations and coordinating collaboration
Translator	Understands the terminology and language of business and technical stakeholders and facilitates communication between the two
Champion	Gains the support of senior management in relation to agile and self-organisation within the enterprise
Promoter	Promotes agile with the customer in order to win their involvement and continued collaboration in support of self-organisation
Terminator	Identifies threatening behaviours and misconduct that are not conducive to self-organisation and engages the necessary support (e.g., from senior management) in order to have such members removed from the team

interpreted. In practice the informal, implicit, transient and spontaneous adoption of roles that address specific concerns in the project (e.g., relations with the customer, securing of senior management support or tackling of tensions within the group) has been observed after which they are then quietly dropped again once the challenge has subsided. Six such roles are listed in Table C.7 (Table 4.2).

In teams that are new to agile techniques there is a pronounced reluctance to adopt such roles which must instead be assumed by the DSDM Coach (or in the case of the Co-ordinator and Translator perhaps by a Business Analyst). As teams mature, however, it has been observed that almost everyone in the team can assume any self-organisation role as and when the need arises owing to the high levels of redundancy, communication and collaboration found in teams that have a collective identity towards the project objectives.

The DSDM Coach embodies much of the Mentor role by providing the initial guidance and support necessary for a team to get to grips with the agile approach. This reflects the common experience of new agile practitioners that whilst conceptually simple, Agile is often difficult in practice and requires thorough knowledge of the techniques and their nuances. The Mentor therefore plays a pivotal role in the transition away from vertical specialist roles towards wider and more encompassing ones ensuring that team members do not misconstrue this metamorphosis as a personal criticism of their abilities or skills. The establishment of an environment of trust wherein opinions and concerns can be expressed without fear of reprisal and the building of confidence within the team so that feedback is borne constructively belong to the people centric concerns of the Mentor as does promotion of self-organising practices (e.g., estimation, planning) in general.

4.5 Products

The Agile Project Framework included a substantial reworking and simplification of products as described in the Atern® and Agile Project Management literature. These include the dropping of the Outline Plan (reducing the levels of planning from three to two), the simplification of the Solutions Foundations by removing the Business Area Definition and Solution Prototype (now subsumed under the Evolving Solution) and redefining the System Architecture Definition, removal of the Delivery Control Pack (e.g., Change Control Records, Communications Log, Issues Log and Risk Log), a slimlined approach to the Solution Assurance Pack (i.e., Solution Review Records, Business and Technical Testing Packs and Deployment Plan that was distinct from the Delivery Plan) and a consolidation of benefits reviews (i.e., Benefits Realisation Plan, Increment Review, Benefits Enablement Summary and End of Project Assessment) into the current Project Review Report and Benefits Assessment.

The DSDM Agile Project Framework now defines a number of products that can be considered project management artefacts in the sense that they are not necessarily part of the final deliverable but constitute supporting documents to ensure that the project progresses smoothly (e.g., project approach and planning, risk management, quality and testing). It is to be expected that the project context (e.g., contractual arrangements, corporate standards and culture) determines how many of these artefacts are to be used and what their necessary level of detail is. Figure 4.11 provides an overview of all these artefacts together with the solution itself.

A distinction is made between evolutionary products (i.e., those that span many phases) and milestone products (i.e., phase specific and of a checkpoint character). Milestone products are best suited for governance purpose and indeed this appears to be their defining characteristic. This distinction, whilst helpful, can seem a little

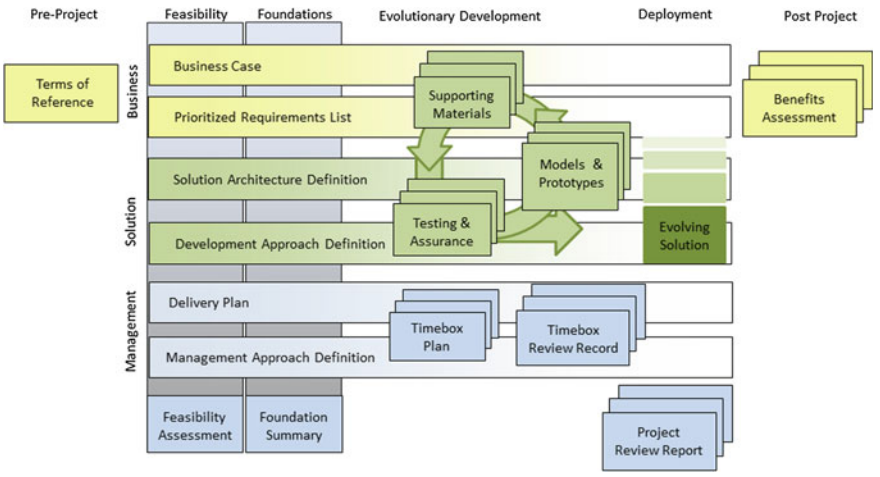

Fig. 4.11 Project Level Products. Published with kind permission of © Alan Moran 2015

Table 4.3 Classification of DSDM Project Products. Adapted from [73]

Category	Products
Evolutionary	Business Case, prioritised Requirements List, Solution Architecture Definition, Development Approach Definition, Delivery Plan, Management Approach Definition, Evolving Solution, Timebox Plan, Timebox Review Record
Milestone	Terms of Reference, Feasibility Assessment, Foundation Summary, Project Review Report, Benefits Assessment

artificial since some artefacts can be said to be both evolutionary and have a milestone nature. For example, both the Timebox Review Record and Project Review Report are intended to record the state of affairs as specific points in the project but evolve by adding additional checkpoints as time progresses (Table 4.3).

An apparently surprising omission is the explicit mention of the Project Approach Questionnaire (PAQ)[14] in the standard product set [73]. This document is a tool used to assess project approach risk in a variety of situations such as where the project environment is not ideally suited to the DSDM approach, where project specific tailoring may be called for or simply as a general means of assessing the readiness of the project team. Based on the outcomes of the PAQ, measures can be introduced to tackle any disparities and to tailor the project approach. The PAQ is, however, considered during the Feasibility and Foundations phases and is generally subsumed under the Management Approach Definition. The following is a brief description of the products recommended by the Agile Project Framework as depicted in Fig. 4.11.

- *Terms of Reference*. This outlines the rationale for the project (e.g., business drivers) and its high-level objectives. In addition key costs, timescales, constraints, dependencies, assumptions and risks ought to be mentioned. It is likely to be a very short document (e.g., one or two pages) that addresses the needs of governance (e.g., statement of strategic alignment, prioritisation of projects within portfolios).
- *Business Case*. This document contains the business vision and justification for the project and requires revalidation throughout the project. The Business Case ought to quantify the costs and value of a project and to explain how this value is delivered on an incremental basis and how this is likely to impact existing business processes and organisation. It is also likely to describe constraints, assumptions, dependencies and risks in the context of a cost-benefit analysis or similar setting that takes the form of an investment appraisal.
- *Prioritised Requirements List (PRL)*.[15] This is the list of all requirements prioritised based on business needs as derived during Feasibility and Foundations (at which point it is baselined meaning that changes to high-level requirements and scope require formal approval). During Timeboxes, items are taken from this

[14] The DSDM Agile Project Framework dedicates an entire chapter and an appendix to a detailed discussion of the Project Approach Questionnaire.

[15] The Prioritised Requirements List mouthful can be abbreviated to PRL (pronounced "prill").

list and elaborated in further detail. Requirements are understood as user needs expressed in terms of services, features or functions that reflect both fitness for purpose (i.e., functional) and fitness for use (i.e., non-functional) and may be formulated as user stories [265]. Requirements formulated during Feasibility have the character of epics (i.e., outcome based high-level statements that clarify scope) and become part of the PRL during the Foundations phase where the first key non-functionals make their appearance. At the Timebox level only those items taken from the PRL are elaborated in fine detail and acted upon. Whilst the prioritised Requirements List is similar to a Scrum Product Backlog, they are nonetheless different. For example, Scrum does not advocate baselining and encourages a forced ranking approach. Both measures reflect the strong product development focus of Scrum rather than one that takes account of project management and governance.

- *Solution Architecture Definition*. This artefact defines the high-level solution design from both business and technical viewpoints. Its primary purpose is to establish the scope for evolutionary development including desired maintainability levels. It is likely to include an overview of current business processes and organisation along with an assessment of the impact of the project on the status quo (e.g., which processes need to be adapted or replaced). From the technical viewpoint the architecture and topology of the solution (i.e., layout of components, choice of technology, technical approach) along with migration considerations (i.e., moving from the old solution to the new one) are likely to feature strongly. The proposed solution should be assessed for feasibility and fitness for purpose which naturally implies that non-functional aspects (e.g., maintainability, quality, security, scalability, performance, usability) must be considered and appropriate measures with which to assess them later be devised.
- *Development Approach Definition*. This provides an overview of tools, practices and standards that are to be adopted in the project (which might reduce to a reaffirmation to use existing standardised approaches) and how these interface with systems and processes outside of the project. Quality assurance measures are also present as is the test management strategy.
- *Delivery Plan*. This is a high-level schedule of project increments together with the planning for the initial Timebox. It is expected that the Timebox Plan for each Timebox elaborates in further detail what is expected of that timeframe.
- *Management Approach Definition*. This artefact considers the organisational and planning aspects of the project as well as the engagement of stakeholders. It is reasonable to expect a stable baseline arising out of the Foundations phase that need only be adjusted as circumstances dictate. At a minimum there should be details concerning the project organisation including suppliers and sourcing model (e.g., outsourcing of specialist functions), assignment of roles, escalation policy and governance model.
- *Feasibility Assessment*. This records the state of affairs across the business, solution and management artefacts as they stood at the end of the Feasibility phase. This serves as an assessment of how feasible the project is likely to be and is either the baseline or executive summary of the documents as they existed at that point in time. It ought to comprise of a statement of the scope and objectives (perhaps

derived or elaborated from the Terms of Reference) in a manner that enables the feasibility assessment of the proposed solution as well as citing any fundamental assumptions (e.g., concerning the state of the market or the strategic direction of the technology being employed for the solution) and risk (e.g., activity of competitors, uncertainties surrounding the solution viability or scalability). Ultimately it must take a view on how likely the proposed solution is going to satisfy the needs of business and though it may explain the exclusion of other alternatives (e.g., on the basis on assessment of risks or costs) it ought not present a "beauty contest" that does not present a clear direction for the project.

- *Foundation Summary*. This artefact plays an analogous role for the Foundations phase as the Feasibility Assessment does for the Feasibility phase though the focus here is more on project appraisal (e.g., it if is likely to result in a return on investment). Accordingly, it is in a considerably better position to make a quantified judgement concerning the project (e.g., in investment appraisal terms). It would seem appropriate to present this artefact as an executive summary and to rely on baselined copies of the underlying products to address follow-up enquiries or clarifications. In environments where commercial considerations beyond costs do not dominate (e.g., Government, NGOs) this summary must decide if the project constitutes a viable and productive use of resources towards the goals of the organisation (e.g., consideration of opportunity cost, assessing of the balance of the political needs of stakeholders).
- *Evolving Solution*. This comprises not only of the solution (be it partial or complete) but also the supporting artefacts produced during its creation (e.g., models, prototypes, tests, reviews). It is desirable that this be a self-contained (albeit be partial and evolutionary) solution that would be capable of servicing some (or all) business needs.
- *Timebox Plan*. This plan elaborates in greater detail those elements of the Prioritised Requirements List as defined by the schedule found in the Delivery Plan that are to be tackled during the Timebox. In a structured Timebox there is time allocated for further clarification of the requirements and thus this plan ought to be of sufficient detail to enable work at the Timebox level to progress. The Timebox Plan should record acceptance criteria, MoSCoW prioritisation and any further relevant information. In pull-based systems (which use the Team Board as the basis of planning) it is not necessary to assign individuals to items on the Timebox Plan.
- *Timebox Review Record*. This is a running review of what has been achieved, feedback, outstanding issues and key decisions and could be constructed to be used for auditing purposes if necessary. The structure of the Timebox Review Record necessarily reflects the Timebox structure (e.g., inclusion of Investigation, Consolidation and Refinement review points). In practice the record may be implemented either as additional attributes on the Timebox Plan or as a separate document.
- *Project Review Report*. Used to capture feedback from each increment, this document records what has been delivered and any business benefits that have been accrued together with feedback at the increment level concerning the process and its practices. Iteratively gathering information in this manner makes for a sound

overall project review once it is concluded. This is a protocol of key decisions, events and formal acceptance of deliverables that forms an excellent basis for a project audit and is likely to be of value both to steering committees and to any reflective activity that looks back on past success and failures of projects (e.g., where a project office or quality assurance unit is attempting to establish and institutionalise best practices).

- *Benefits Assessment*. This product reflects on the accrued benefits once the solution has been in use over a period of time. The precise composition of this assessment is left rather open but it seems conceivable that it either follows a similar format to the Project Review Report or be manifest as several documents that review benefits over ever longer timeframes. It formally notes what was delivered and what was omitted (based on the contents of the Project Review Report) and may form part of project decommissioning activities. Where possible it should endeavour to quantify benefits (e.g., analogous to project appraisal techniques) and be capable of analysing discrepancies between the predicted and actual outcomes. It should also have sufficient depth to assess the promised coverage of benefits found in the Business Case (using the success criteria stated therein) and differentiate or otherwise account for the contribution of the Deployed Solution (e.g., mitigating for factors arising from the changing business environment). This is the only document produced by the Business Visionary (albeit with the analytical assistance of the Business Analyst) and thus ought to reflect mostly a business perspective on the final outcome of the project.

Whilst this is a comprehensive list of products there is no compulsion to require all of them or suggestion that they are complete for all project needs. For example, in small projects it is not uncommon to merge the Feasibility and Foundations phases which in turn implies that the Feasibility Assessment and Foundations Summary would coalesce into a single document.

The DSDM agile chart of products, illustrated in Fig. 4.12, shows how products map to the distinct levels of project (where the Feasibility and Foundations and later Deployment phases have been subsumed), increment and Timebox. Two natural slices clearly emerge marking the different levels of planning (guided by the PRL and Delivery Plan which are detailed further in the Timebox Plan) and review (i.e., Timebox Review Records, Project Review Report and Benefits Assessment). Other elements (e.g., Business Case and definition and approach documentation have their epicentre at the project level but may be influenced by events at the increment level).

Finally, Fig. 4.13 describes the roles and responsibilities in relation to the products as derived from the DSDM Agile Project Framework in terms of RACI[16] chart. What becomes evident from is that most of the accountability is focused in the roles of Business Sponsor, Business Visionary, Technical Co-ordinator and to a lesser extent Project Manager who together also bear much of the responsibility for DSDM products. Accordingly the composition of this group is important. For example, the combination of Business Sponsor and Project Manager would give rise to a number

[16] RACI (Responsible, Accountable, Consulted and Informed) describes the manner in which roles are engaged and is a common device in process management.

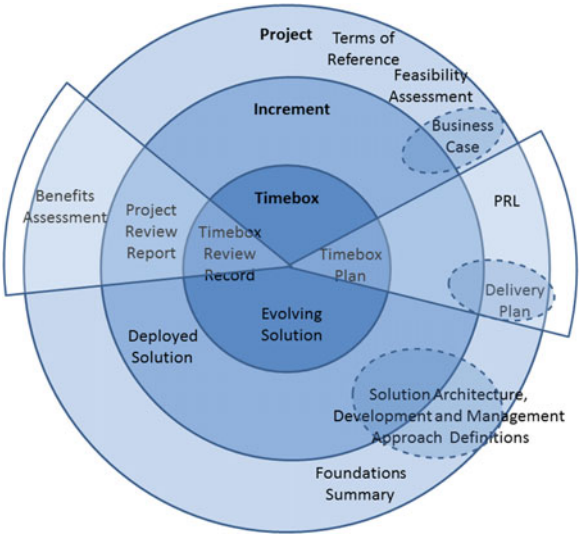

Fig. 4.12 Agile Chart of DSDM Products. Published with kind permission of © Alan Moran 2015

	Terms of Reference	Business Case	Prirotised Requirements List	Solution Architcture Definition	Development Approach Definition	Delivery Plan	Management Approach Definition	Feasibility Assessment	Foundations Summary	Evolving Solution	Timebox Plan	Timebox Review Record	Project review Report	Benefits Assessment
Business Sponsor	A	A				I	A	A	A	I		I	I	A
Busines Visionary		C	A	A		A C	C	C	C	A	A	A	A	R
Business Advisor		(C)	C	C	C	C	I	C	C	C	C	C	C	
Technical Coordinator	I	C		R	R	A C	C	C	C	A	A	A	A	
Technical Advisor		(C)		C	C	C	I	C	C	C	C	C	C	
Project Manager		C		A	A	R	R	R	R	I	A	I	R	
Business Analyst	I	R	R	R		C	I	C	C	C	I	C	C	R
Business Ambassador*			C											
Solution Development Team		I	I	C	I	C	I	C	C	R	R I	C	C	
Team Leader*												R	A	
DSDM Coach					C		C	C	C	C			C	
Project Governance Authority	I	I			I	I	I	I	I			I	I	I
Anybody	R C									I				C

* Where different to collective responsibility of the Solution Development Team

Fig. 4.13 RACI Chart for DSDM Roles and Products © DSDM Consortium 2014. Reprinted with permission

of potential conflicts of interest owing to the conflating of of accountability and responsibility of several products. Approvals must therefore remain balanced evenly between the Business Sponsor (who focuses on high-level solution purpose and definition as well as approach matters) and the Business Visionary and Technical Co-ordinator (who share accountability at lower levels of detail) though the Project Manager occasionally approves definition and planning artefacts.

4.6 Planning and Control

As suggested by Fig. 4.12, DSDM incorporates broadly two levels of planning, the Delivery Plan and the Timebox Plan.[17] The highest level of planning is the agreement of strategy and focuses on solution delivery and quality assurance by outlining increments, their timeboxes and how review and testing activity is to be incorporated throughout. Generally speaking this is concluded in outline form during the Feasibility and Foundations phases during which time the Delivery Plan has been initiated. During the Evolutionary Development phase further detail is captured in the Timebox Plans as a result of the refinement of the Delivery Plan that occurs mostly at the project and increment levels. Arguably there is a third level of planning implied by the planning slice that includes the Daily Stand-Up (see Fig. 1.14 of Chap. 1) wherein each day commences with a discussion of what has been achieved since the last stand-up, what is planned until the next meeting and what impediments exist. In terms of timelines a Delivery Plan will project a planning horizon of several months whereas a Timebox Plan focuses on activities over the coming fortnight to weeks (and of course the Daily Stand-Up that concentrates on the coming hours of work). The act of transitioning a deliverable into use, referred to as deployment, requires refinement of the Delivery Plan with an update of whatever deployment activities are necessary.[18] On delivery of each increment, however, it is advisable to revisit the details of the Foundations phase to assess the continued viability of the project and to embed any relevant findings into the planning of the next increment.

The impact of the dual levels of planning on requirements is that the estimation accuracy grows in tandem with the level of detail of requirements thereby balancing the need to avoid being held hostage to unreliable estimates against the tendency to dive too deep into details too early on. The role of prioritisation also needs to be born in mind since it may be the case that the resource and budget constraints are insufficient to cover all requirements. Accordingly, their categorisation in terms of MoSCoW ensures smooth management of requirements between the two planning levels (e.g., a requirement marked as “W” at the Timebox level leaves open the possibility of

[17] Earlier DSDM literature cited three levels of planning that also included a high-level Outline Plan though this has since been subsumbed in the Foundations activities in the DSDM Agile Project Framework.

[18] In earlier DSDM literature the Delivery and Deployment Plans were considered separate products with the latter being considered an evolution of the former.

it being taken up in another Timebox by which point in time its priority may well have been promoted). The nature of planning in DSDM enables confidence in the evolving solution to be built up over time. This is important in light of the systems thinking remarks in Chap. 1 where attention is drawn to the dynamics of trust, quality and timeliness of feedback and the perceived performance of the solution which ultimately impacts the rate of change of business satisfaction (see Figure 1.17).

4.7 Management Implications

DSDM was the first agile methodology with a clear project management focus[19] and though only its core elements were introduced in this chapter it will be repeatedly referred to when discussion turns to matters of governance, quality and risk management. Accordingly it presents the most complete option for managing projects in a manner consistent with agile principles and compatible with best practices at the programme level. At the operational level, the fact that DSDM shares practices with Scrum and related methodologies (e.g., Free Format Timebox, Daily Stand-Up, User Story formulation of requirements) enables such approaches to be embedded within its process thereby marrying review and control structures required by senior management with elements of solution development with which practitioners feel most comfortable.

The DSDM roles provide a framework for identifying responsibilities that must be assumed by the team as well as identifying possible conflicts that might otherwise have gone unnoticed. Though many of these roles may appear to be familiar to those with a more traditional background, a genuine risk lies therein of them being subordinated to their non-agile cousins. For example, a project manager may need to be significantly reinterpreted along agile principles (if this is not already the case) and the role of Team Leader may be neither permanent nor hierarchical in nature. To an extent devices such as Fig. 4.13 may assist in the localisation of conflicts between the roles. Realignment with new responsibilities and the related change in mindset remains one of the greatest challenges for personnel transitioning to more agile approaches. If these explicit roles reflect the static structure of the team, then the implicit roles arising from self-organisation capture its dynamic. Moreover, the readiness to assume (and relinquish) these roles in light of the coming (and passing) of challenges facing the team is a reflection of collective responsibility and agile maturity. A corollary of this, however, is that much of the dynamic responsibility within team that is new to Agile might need to be assumed by a DSDM Coach until such time as the team is ready to take on these responsibilities. Knowing that they exist, preparing the team to become comfortable with them and building their confidence in engaging issues therefore belong to the remit of the coach. Furthermore,

[19] In recent years other methodologies (e.g., DAD, SAFe®) have incorporated elements of project management to varying degrees (e.g., SAFe® avoids using the term project preferring instead to refer to team level structures).

since the guardians of methodological practice and team building and organisation are likely to be the support staff (e.g., DSDM Coaches, Workshop Facilitators), it follows that establishing a support infrastructure (e.g., agile PMO) with such capabilities would make sense for the organisation. Organisational learning will inevitably have an impact on the fluidity of roles as team members exchange and share information which in turn promotes their own development of "T-shaped skills". This breakdown of functional boundaries and hierarchical structures (promoted by self-organisation, internal autonomy and team flexibility) is to be encouraged and thus it helps not to be overly rigid in either the assignment of roles or their interpretation.

The overview of DSDM products described in this chapter will be returned throughout this book and their distribution across the phases (see Fig. 4.11) will be used to pin point their relevance in addressing wider project management concerns (e.g., quality and test management, risk management). A feature of this product set, also shared by the planning and control practices, is the progressive emergence of detail and the manner in which decisions may be permitted to be delayed until such time as sufficient information exists. Thus it is adaptation rather than omission that characterises the approach and it is feedback rather than rigidity that steers it forward.

Chapter 5
Agile Programme Management

Abstract Typically an organisation is involved in multiple undertakings some of which can be aggregated together to serve a common vision whilst others represent disparate or opportunistic activities. Combined they enable the transition of an organisation from its current state to a desired future state which inevitably gives rise to tensions between tactical (i.e., operational) and strategic concerns which must also be managed alongside wider organisational change. Change itself may be triggered by market movements, new regulatory terms of reference or a multitude of other reasons resulting in a continual sense of transformation and transition within the organisation. A programme represents one of the highest levels of change management wherein multiple related projects are combined in order to deliver new capabilities. There is a strong similarity between the DSDM Agile Programme Framework and Managing Successful Programmes® both of which use the same terminology and broadly similar structures and products suggesting that the former is an evolutionary idea nuanced with some agile details (e.g., delegation of decision-making, incremental delivery of capabilities) rather than a seismic challenge to existing thinking on programme delivery.

5.1 Introduction

Classical change initiatives are classified in terms of portfolios, programmes and projects. This may be derived from an overarching strategic portfolio management a subset of which is the subject of active portfolio management at any given point in time as suggested by Fig. 5.1. Portfolios are the highest level of change and embody the initiatives that an organisation is undertaking or planning to undertake which can be disparate and unrelated in nature. It is at this level that an organisation must ultimately define and justify itself and how it intends to create a return on its investments (in the private sector) or act on its political mandate (in the public sector). Programme management is concerned with the optimal resourcing of multiple related projects in a manner that is consistent with the overriding corporate strategy. Its function is to encompass governance, oversight for processes and methodologies (including their continual improvement and optimisation) as well as provide support. A programme

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_5

is considered a temporary arrangement that is dissolved once its outcomes have been achieved wherein individual projects are the vehicles of change tasked with the delivery of specific outputs (e.g., creation of a software solution) for which resources (e.g., staff, finances, suppliers) have been assigned. Projects (as discussed in Chap. 4 are thus the lowest unit of change vehicle and concern temporary organisational constructs that deliver specific outputs that ultimately contribute in part to the new emerging capabilities of an organisation. Ostensibly the DSDM Agile Programme Framework does not mandate any specific agile method at the project level though it is clear that DSDM-phile methodologies are preferred. Here, a capability refers to the delivered benefits (as perceived by customers and stakeholders) typically expressed in terms of business processes, people, physical assets and information. Capabilities arise when the outputs of one or more projects, grouped together as a tranche within a programme, are combined to deliver the benefits as indicated by Fig. 5.2.

These supra-project structures defined at the enterprise level connect strategy to change activity and bring with them complex support functions and management concerns revolving around the role of governance and risk management as they apply to the organisation as a whole. Whilst it is possible to engage these matters in an agile manner (e.g., placing emphasis on incremental deliver consistent with emergent strategies, incorporating feedback and review to enable an iterative approach and promoting organisational learning, cultivating a culture of empowerment and ensuring malleable self-organisational structures) the reality is that at this level there will often be a mixture of agile and non-agile approaches. This is to be expected since although

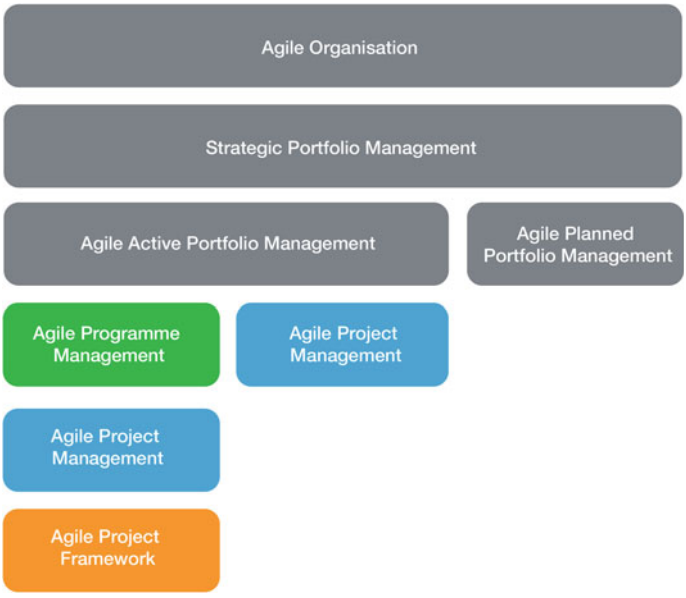

Fig. 5.1 The Agile Organisation © DSDM Consortium 2014. Reprinted with permission

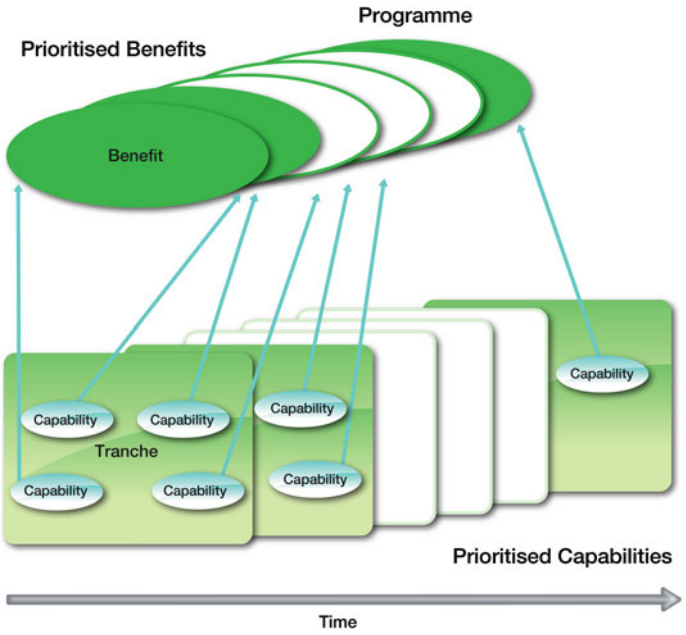

Fig. 5.2 Relationship Between Benefits and Tranches © DSDM Consortium 2014. Reprinted with permission

an organisation may well have achieved a high degree of agility within its own borders, it will find itself interacting with partners, suppliers and other participants in its value chain that have adopted different approaches on the delivery of change.

In agile terms programmes are expected to remain aligned with current business strategy and in particular to exhibit the features necessary to cope with the changing direction of emergent strategies (as described in Chap. 2). For example, this is achieved through the continual sensing of and adapting to the business environment and the iterative development and incremental delivery of outcomes focusing only on what is needed at any given point in time. As discussed in Chap. 6 appropriate governance structures must accompany the programme though it is already evident at this point that top-down external project governance models are likely going to be favoured. Thus whilst active stakeholder involvement is encouraged, the DSDM Agile Programme Management Framework [72] appears to hold back from endorsing open democratised governance models. The following principles apply to DSDM at the programme level:

- *Programme goals are clearly and continuous aligned to business strategy*. This principle mandates frequent review and validation of the programme (e.g., when there are changes to business strategy, capabilities are enabled or on conclusion of projects). In spite of the appearance of the assumption of an implicit planned strategy being the driver with less accommodation of emergent strategic thinking,

there is sufficient mention of adaptation elsewhere (albeit at different levels) to warrant the belief that change can be effected in both directions. Review along with continued sponsorship and commitment towards the Business Case are necessary elements of this principle.

- *Benefits are realised incrementally and as early as possible*. Early and incremental delivery of benefits implies partial delivery of benefits. This admits the possibility of a change of course based on new information thereby negating the necessity (in the short term) to plan in detail those benefits that will only be realised in the long-term. The prioritisation of programme benefits must of course trickle down to the prioritisation of capabilities delivered by projects which must take into consideration cost-benefit factors.
- *Governance focuses on creating a coherent capability*. Issues of programme driven change, synchronicity of capability delivery and coherence of the overall programme are complex matters that cannot be governed at the project level. This require high-level planning and an undertaking to understand where projects overlap and what their inter-dependencies may be. The role of governance is therefore to ensure that integrated and coherent capabilities are aligned with the overall vision are realised.
- *Decision-making powers are delegated to the lowest possible level*. This principle is concerned with achieving the right balance of programme led decision-making and governance whilst ensuring internal autonomy at the project level.
- *Agile programmes are iterative and have the ability to contain both agile and non-agile projects*. This is a pragmatic recognition of the diversity found within large programmes and the need to accommodate non-agile approaches whilst adhering at the programme level to agile principles.

5.2 Lifecycle

Incremental delivery at the programme level is encapsulated in the notion of a programme increment also referred to as a tranche. Thus the spectrum of enablement activities involves one or more capabilities linked to individual projects which between them are delivered and enabled incrementally through tranches as depicted in Fig. 5.3. The implications that this has for planning is that planning horizons are effectively terminated at each tranche boundary though the overall vision is expected to extend further.

The programme lifecycle contains the following six phases that bear a resemblance to the DSDM project model as described in Chap. 4 together with an overarching Benefits Management process as illustrated in Fig. 5.4.

- *Pre-Programme*. This phase derives a programme from the existing portfolio within the organisation and sets about determining the initial key roles to be involved (e.g., Business Programme Owner, Business Change Owners). It is here

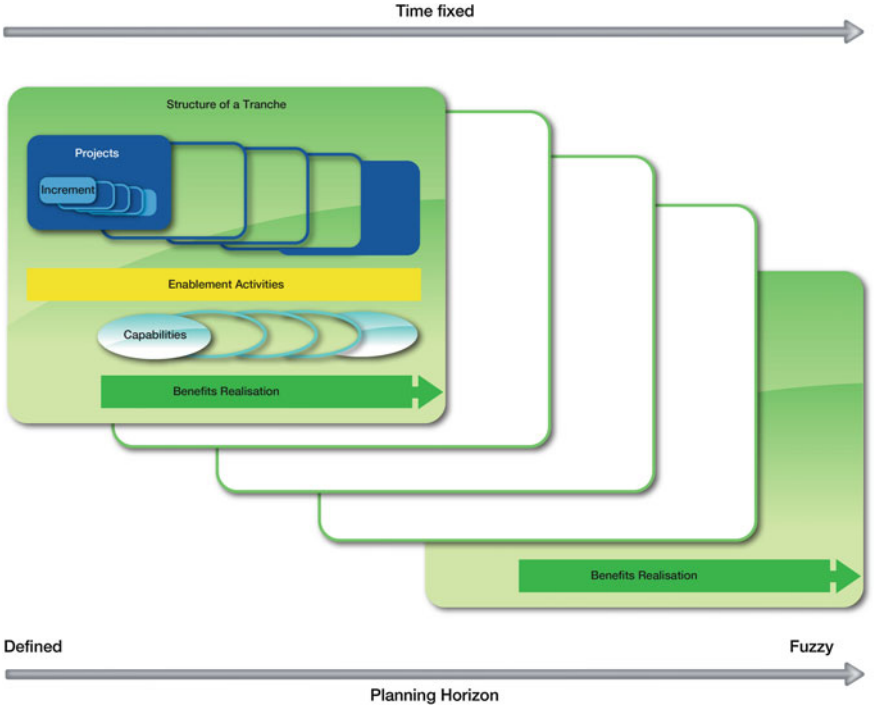

Fig. 5.3 The DSDM Agile Programme Structure © DSDM Consortium 2014. Reprinted with permission

that the decision is taken if the programme is worth pursuing though if so, then planning is deferred to later phases.

- *Programme Feasibility*. The inclusion of the programme in the active portfolio management is a decision that is based on cost-benefit analysis, strategic alignment and consistency, clarity surrounding the nature of the transformation, its approach and governance and the readiness of the organisation to absorb it.
- *Programme Foundations*. The basis on which the programme rests must be clarified in terms of the business architecture, intended benefits, management structure and governance, stakeholder engagement policy and communication and high-level plans. In contrast to traditional programme management the detailed planning of tranches and individual projects is deferred until sufficient information is present. Notwithstanding, the first tranche can be planned insofar as project and the capabilities to which they contribute can be identified and their outline budgets, dependencies and timelines can be determined. At this point various artefacts should come into focus including the Business Architecture Model and Roadmap, the programme Business Case and Programme Plan and related supporting materials.
- *Capability Evolution*. At the tranche level specific capabilities are expected to be iteratively developed and incrementally delivered subject to continual review

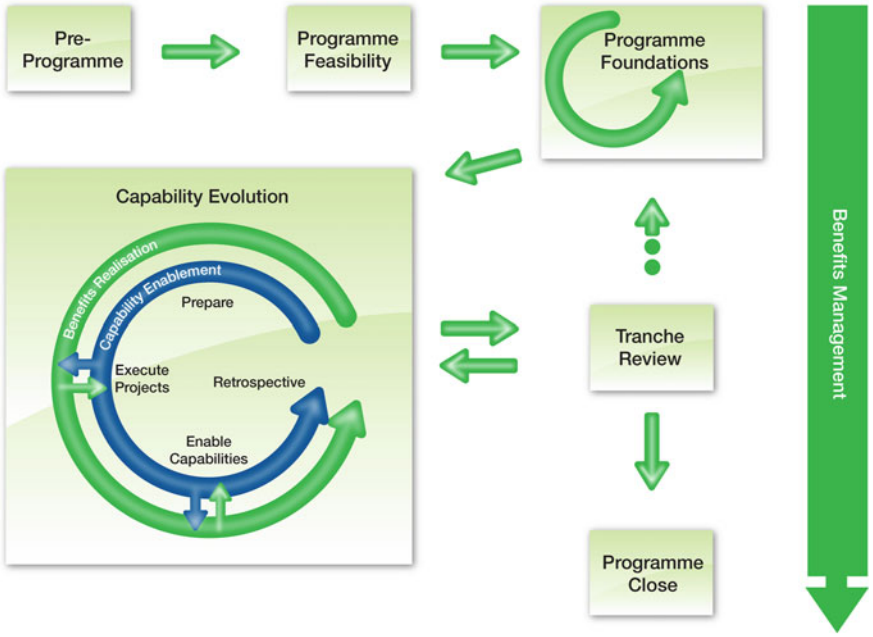

Fig. 5.4 The DSDM Agile Programme Lifecycle © DSDM Consortium 2014. Reprinted with permission

and assurance of convergence towards the desired future state. Note that tranches may overlap provided that they are self-consistent and are not dependent on each other. Furthermore a tranche does not need to deliver a full capability as it suffices for it to contribute part functionality. Indeed, this is desirable if learning is to be fed back into the overall programme. During this phase capabilities are enabled which requires that stakeholders are on board, that projects are executed and that their deliverables are assembled so that existing processes can be changed or adapted to make use of them allowing time at the end for reflection on the process. Finally benefits realisation benchmarks assesses capability enablement to see if partial benefits are being realised even if the entire capability is not yet present. This also allows for learning to influence the evolution of capabilities that will be incrementally built upon in the following tranches.

- *Tranche Review*. The primary purpose of this phase is to assess the on-going viability of the programme. Reviews may commence as soon as sufficient capability has been enabled and though parallel tranches ought not to influence each other there may still be collateral considerations to take into account should one review call into question matters that impact the overall business architecture or planning. Tranche reviews will determine when the tranche (and ultimately programme) is considered done and will trigger the initial planning of subsequent tranches.
- *Programme Close*. Closure requires merely the confirmation that sufficient capability has been delivered with a review against the vision and business case

along with a collection of lessons learned. From an agile perspective, detection of divergence between planned and realised benefits may not be as major a concern as it might seem since this might simply reflect the degree of adaptation to changing circumstances. Furthermore, lessons gathered at this stage could be said to have relatively little potency since there is no longer an internal programme feedback loop. Moreover the transfer of such lessons to other programme contexts might be prove to be of relatively little relevance.

Benefits management, which begins during Feasibility and Foundations and is built upon iteratively thereafter, tackles benefits realised as soon as a capability becomes enabled. This process identifies, monitors, measures and ensures continued on-going commitment to their realisation. Accordingly it must operate in tandem with the underlying cadence of tranches and incremental delivery of capabilities.

Throughout communication plays an important role since stakeholders who must come to terms with the change being introduced may offer resistance if not sufficiently convinced of the benefits. Accordingly incremental enablement of capabilities and delivery of benefits serves to build trust. True communication (rather than mere informing) ensures that valuable feedback is gathered and fed back into the programme. Thus communications encompasses both outward and inward looking aspects. The apparent weight of communication (e.g., Stakeholder Engagement Strategy and Communication Plans at the tranche level) is perhaps a reflection of the complexity that it assumes at this level when compared to the more integrated communication found at the project level.

5.3 Roles and Responsibilities

Programme management roles invariably entail people development and leadership skills (e.g., trust and respect in team members and the ability to motivate and inspire), a degree of business acumen (e.g., ability to prioritise and take calculated risks) and strategic foresight (e.g., the ability to determine and communicate the programme vision). As is the case with project roles three classifications (i.e., business, technical and managerial) exist and an individual may hold more than one role or a role may be split across multiple individuals. These roles, the details of which can be found in Appendix C, are listed in Table 5.1 and depicted in Fig. 5.5.

The Programme Management Team comprises of roles that for the most part are involved both in business strategy and programme management and are likely to be a suitable pool for programme and project steering committees owing to the authority and access to resources within the organisation that they command. The are accountable for oversight and must direct and strategically guide the transition of the organisation towards its new future state that the programme is attempting to achieve. They are supported in administrative affairs by the Programme Support Office and must have the necessary respect of the network of key stakeholders and corporate managers linked to the programme.

Table 5.1 Programme Role Matrix

Interests			
	Managerial	Business	Solution/technical
Programme Management Team	Business Programme Owner, Business Change Owner, Programme Manager	Business Programme Owner, Business Change Owner	
Capability Delivery Team	Stakeholder Engagement Co-ordinator	Programme Change Architect, Project Team	Programme Technical Architect, Programme Change Architect, Project Team
Supporting	Programme Support Office	Change Agents, Specialists	Specialists

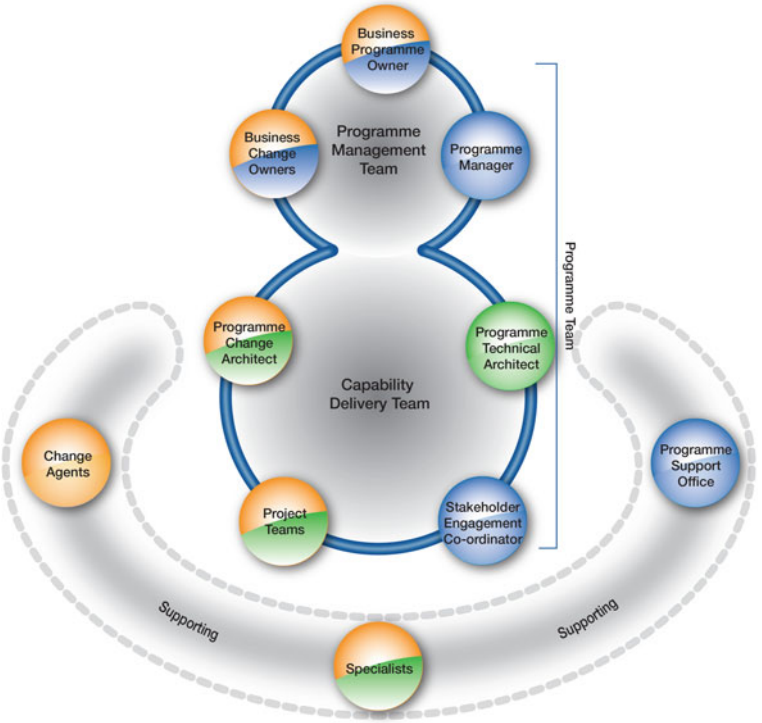

Fig. 5.5 Agile Programme Framework Roles © DSDM Consortium 2014. Reprinted with permission

Capability delivery roles subsume not only individual project roles but also include technical architectural individuals who must ensure consistency of design across the programme (e.g., by becoming involved in matters that can not be resolved as effectively at the project level) as well as being accountable for stakeholder engagement, management and communication. The existence of these roles is necessary if other-

wise autonomous projects are to pull together effectively in the right direction and are thus crucial for the enablement of capabilities.

Supporting roles fulfil a variety of support functions and their organisation is a relatively situational affair that depends on the nature of the organisation (e.g., use of contractors, existence of internal support infrastructures). Though not explicitly mentioned (or perhaps subsumed under Specialists) there may also be methodological experts who can support the added complexity of running a programme in an agile manner.

The interface between programme and project levels is illustrated in Fig. 5.6 where possible points are contact are indicated. Thus the parallels between Programme

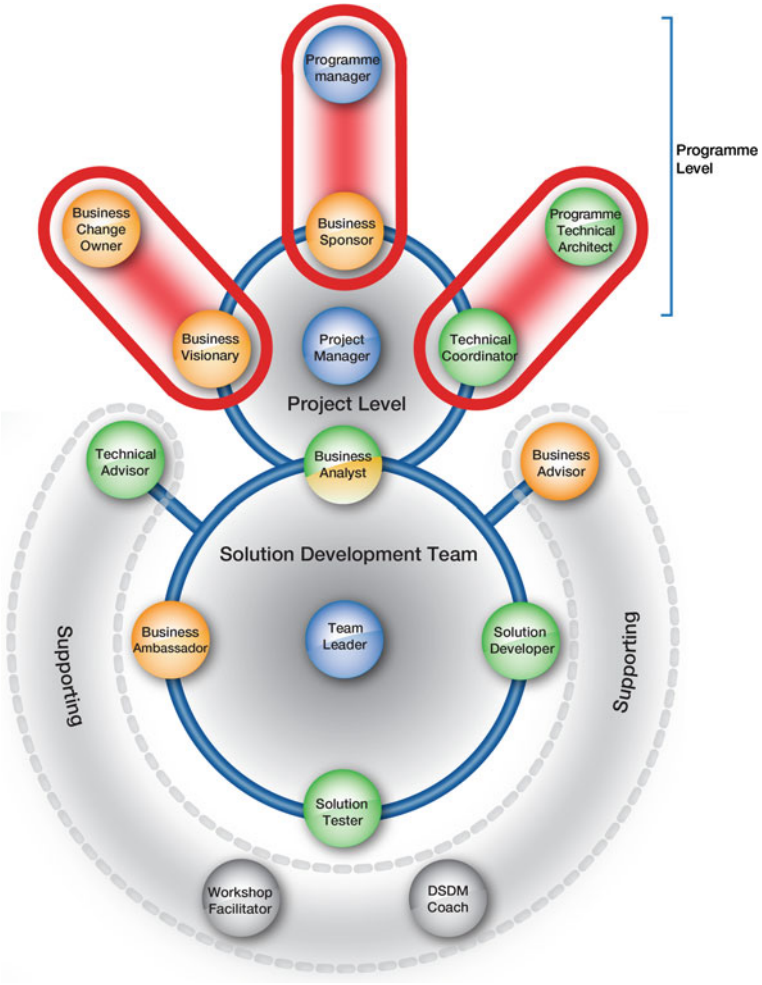

Fig. 5.6 Interface Between Agile Programme and Project Roles © DSDM Consortium 2014. Reprinted with permission

Technical Architect and Technical Co-ordinator are such that they may be identified with each other suggesting close collaboration between these roles is to be expected. On the other hand the assumption of the Business Sponsor role may well be done by the Programme Manager[1] though the Business Change Owner, who could equally assume the role of Business Visionary, may be more appropriate. Irrespective of the details, however, there should be a single Business Sponsor and Business Visionary at the project level where possible and the precise mapping of programme to project must of course take account of their individual capacity, ability to commit and appropriateness of match (e.g., business area compatibility).

5.4 Products

The DSDM Agile Programme Management products (see Table 5.2 and Fig. 5.7) can be categorised as evolutionary, milestone or foundation artefacts that address business, capabilities and benefits, benefits management and programme management interests. Interestingly the rather long list of products required at the programme level bears some resemblance to the list of the DSDM Agile Project Management[2] artefacts suggesting that the intention is that they be used together. Furthermore, one might expect there to be downward pressure on projects to integrate themselves into the programme document structure (e.g., input into high-level project status reporting rather than the local use of Team Boards). However, a well-governed project (e.g., one that employs Timebox Review Records) ought not need to produce any additional reporting for the programme and thus the artefacts that it already uses (e.g., Burndowns, Team Boards) may remain in place provided that the project can report back on progress relating to its Minimum Useable Subset.

One may still be forgiven for asking why this approach should be considered agile in light of the preference for working solutions over comprehensive documentation and the principle of conveyance of information through face-to-face communication expressed in the agile manifesto. An adequate answer to that question must take into account how practices at the programme level are enacted (e.g., depth and scope of documentation) and how agile principles manifest themselves (e.g., iterative development and incremental delivery of capabilities) whilst also taking into account that a programme necessarily introduces higher levels of complexity in terms of communication and coordination. It is during Programme Foundations that most documents are baselined, though some documents (e.g., Communication Plan, Tranche Plan and Tranche Review Record) are also baselined during the Tranche Review phase for each tranche. Throughout the DSDM literature other artefacts are mentioned that are not included in the core document set. These include the

[1] DSDM defends this assignment in spite of its apparent dissonance suggesting that it is the preferred approach.

[2] Note that what is being referred to here is DSDM Agile Project Management not the DSDM Agile Project Framework.

Table 5.2 Classification of DSDM Programme Products

Category	Products
Evolutionary	Vision Statement, Prioritised Benefits Definition, Business Architecture Model, Roadmap, Benefits Realisation Plan, Business Case, Stakeholder Engagement Strategy, Communication Plan, Programme Risk Log, Programme Issues Log, Communications Log, High-Level Project Status, Benefits Assessment
Milestone	Governance Strategy, Tranche Plan, Tranche Review Record
Baselined	Vision Statement, Prioritised Benefits Definition, Business Architecture Model, Roadmap, Benefits Realisation Plan, Business Case, Governance Strategy, Stakeholder Engagement Strategy, Communication Plan, Tranche Review Record

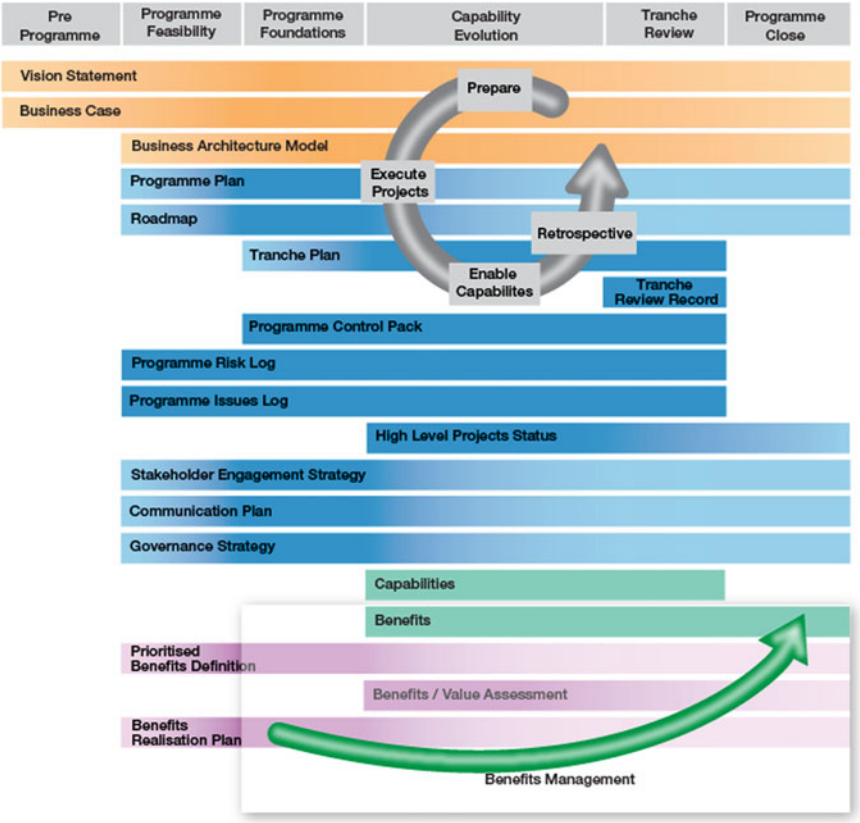

Fig. 5.7 Programme Level Products © DSDM Consortium 2014. Reprinted with permission

Programme Approach Questionnaire and the Configuration Management Strategy though these are alluded to elsewhere in this book where appropriate.

- *Vision Statement*. This document attempts to capture the intended state of the organisation following the transformation programme as reflected in the business strategy. The challenge therein lies in creating something that is understood, meaningful, achievable and can be used as the basis for validating decisions. Created in the early phases of the programme, it may be updated throughout as circumstances dictate.
- *Prioritised Benefits Definition*. This document describes the intended benefits and more importantly how they are to be measured and assessed thereby communicating clearly intentions and expectations to the capability delivery teams. As the programme progresses these benefits will necessarily be fleshed out in more detail. The point of prioritising them (using MoSCoW) is to ensure that the more valuable benefits can be delivered earlier.
- *Business Architecture Model*. Where a vision statement captures state, the Business Architecture Model describes structure of the future organisation (e.g., processes, organisational structures, technology and information). This would ordinarily include a prioritised capabilities list that is linked to the Prioritised Benefits Definition along with a series of incremental states through which the model may progress.
- *Programme Plan*. Containing both the Roadmap and the Benefits Realisation Plan this describes the route to capabilities enablement and realisation of benefits. Ideally it should project high-level consistency and be compatible with related artefacts (e.g., Business Architecture Model).
  - *Roadmap*. This describes the schedule of tranches that make clear the incremental nature of capabilities enablement and ought to cover high-level details of resourcing and costs required to realise the schedule whilst admitting the fact that future tranches bear a higher degree of uncertainty in this regard.
  - *Benefits Realisation Plan*. This document outlines a mapping between accrued benefits as capabilities come online and is linked to the Prioritised Benefits Definition. Inherent in these relations is the fact that whilst some benefits arise gradually with the emergence of new capabilities, others are structurally held back until specific combinations of capabilities are enabled and integrated. The value of this document therefore lies in the fact that such details are clearly communicated to the respective Business Change Owners whilst also identifying the necessary coordination in order for the desired outcome to be reached.
- *Business Case*. This product must detail the risks and rewards bound to the programme as a whole in the context of the wider organisation (e.g., acknowledgement of the impact of the transformation on enterprise risk management). Funding is understandably more complex since the uncertainty regarding future tranches may well undermine estimation efforts and therefore gated funding models at the tranche level are likely to be used. Note that the existence of a programme level Business Case means that individual projects must not formulate their own

Business Cases beyond perhaps contextualising the programme Business Case in terms of their own business area.

- *Governance Strategy*. This strategy defines the governance model to be used within the programme i.e., on what basis decisions in the programme will be made. It is to be expected that the details will cover escalation triggers and paths, roles and responsibilities and any matters concerning reviews or approvals. By implication this frames governance as a project external mechanism (see Chap. 6 for further details) from which individual projects must take their lead and thus it is down to the Project Managers and programme governors to resolve any tensions that may arise as a result.
- *Stakeholder Engagement Strategy*. A strategy to cater for the needs of internal and external stakeholders needs to be formulated and may include a stakeholder analysis and engagement and interaction approach that keeps stakeholders committed to the outcomes of the programme.
- *Communication Plan*. This is a tranche rather than a programme level entity that identify the key events and communications that need to be distributed and the nature of the communication policy that needs to be employed. Consistency and appropriateness of the message is of course important as is the gathering of feedback.
- *Tranche Plan*. These plans represent the second level of programme planning and emerge just prior to the initiation of each programme increment. They detail prioritisation of capabilities and their related projects (though the project details themselves are omitted). This exerts some degree of influence on the Timebox planning of projects since it is at the tranche level that dependencies between projects become apparent. Only the first tranche is planned at the start of the programme with subsequent planning events occurring through the programme in light of the state of information that is then available. Anticipated benefits that should arise from the tranche should correlate with other products (e.g., Benefits Realisation Plan) and straight through thinking should prevail in relation to costs, key resources, impact of new capabilities and their risks.
- *Tranche Review Record*. In the closing stages of completion of a tranche an assessment of whether or not sufficient capability has been delivered needs to be made and any relevant findings integrated into the Programme Plan. Matters relating to formal acceptance of capabilities and the benefits that have been realised together with lessons to be carried forward also belong to this artefact.
- *Programme Control Pack*. This is essentially a status control compilation that serves to identify risks and keep participants informed of progress:
  - *Programme Risk Log*. Corresponding to the traditional risk register this is a top-down assessment of programme level risks and the strategies needed to tackle them. The tendency, also found in early DSDM literature, to frame risk as a purely negative phenomenon makes a reappearance here and there is a recommendation that project level risks not be included unless they have a

programme impact.[3] The Programme Approach Questionnaire, not cited as a separate product, is an important risk related artefact.
  - *Programme Issues Log*. The issues log contains programme level concerns that require management attention with those affecting individual projects not included. There is perhaps an implicit assumption that the DSDM Agile Project Management method is being used since this does indeed contain an Issue Log in its Delivery Control Pack though the product structure of the DSDM Agile Project Management Framework makes mention of neither.[4] Be that as it may, the log must merely record issues, their severity and whether or not they have been addressed in a timely manner.
  - *Communications Log*. This log contains a record of all key communications transmitted by the programme together with relevant decisions taken at that time.
  - *High-Level Project Status*. Classical reporting in terms of progress, expenditure and any risks and issues constitute this aggregate report that gathers information from individual projects. This implies that project level reporting is in place and that this is reasonably standardised across all projects (be they agile or not). For example, agile projects that might otherwise limit reporting to the use of a Team Board would need to gather the necessary information in written report format to pass up to the programme level.
- *Benefits Assessment*. This assessment is essentially the reporting element of the Benefits Realisation Plan and attempts to measure what benefits have up to that point being realised through the incremental delivery of capabilities. Clearly this cannot be adequately captured at the project level and a degree of consolidation at certain key points (e.g., tranche reviews) is to be expected.

Closer analysis of the DSDM Agile Programme Management literature (see Table 5.3) reveals that the roles of Business Programme Owner and to a lesser degree Programme Manager are critical to the success of the programme since between them ownership of most of the artefacts is consolidated. This applies even to artefacts such as the Stakeholder Engagement Strategy and Communication Plan which one might otherwise expect to be owned by the Stakeholder Engagement Co-ordinator.

5.5 Planning and Control

Programmes must anticipate change not only on account of the volatility of the business environment but also on account of their comparatively long horizons (that are usually expected to extend over years). Thus it is reasonable to develop and

[3] This recommendation does appear to be at odds with the common practice of risk aggregate employed by enterprise risk managers to assess whether or not individual but connected risks on different projects do in fact constitute a programme risk.

[4] The confused reader may wish to review again the difference between the various DSDM frameworks found in Chap. 1.

Table 5.3 Product Responsibilities (O = Owner, A = Author/Updater, C = Contributor)

	Business Programme Owner	Business Change Owner	Programme Manager	Programme Change Architect	Programme Technical Architect	Stakeholder Engagement Co-ordinator	Change Agent	Programme Support Office	Specialist	Project Manager
Vision Statement	OA	C	C	C	C	C			C	C
Prioritised Benefits Definition	OA	A	A	C	C	C	A	C	C	C
Business Architecture Model	O	C	C A	A	C			A	C	
Programme Plan Roadmap	A	A	OA	A	A	A			A	A
Benefits Realisation Plan	OA	A	A	A	C	C	C	C	C	C
Business Case	OA	A	A	C	C	C	C	C	C	C
Governance Strategy	OA	A	A	C	C	C	C	C	C	C
Stakeholder Engagement Strategy	OA	A	A	C	C	A	C	C	C	C

Table 5.3 (Continued)

	Business Programme Owner	Business Change Owner	Programme Manager	Programme Change Architect	Programme Technical Architect	Stakeholder Engage-ment Co-ordinator	Change Agent	Programme Support Office	Specialist	Project Manager
Communication Plan	O	A	A	C	C	A	C	C	C	C
Tranche Plan	A	A	OA	A	A	A	C	C	A	A
Tranche Review Record	C	C	OA	C	C	C	C	C	C	A
Programme Control Pack										
Programme Risk Log	C	C	OA	C	C	C	C	C	C	C
Programme Issue Log	C	C	OA	C	C	C	C	C	C	C
Communications Log	C	C	OA	C	C	C	C	C	C	C
High-Level Project Status	C	A	OA	C	A	C	C	C	C	A
Benefits Assessment	O	A	A	C	C	C	C	C	C	C

deliver capabilities incrementally in order to navigate to transition states that reflect an interim but incomplete versions of the overall programme vision. Tranches are focused on the medium term future (e.g., a few months to a year) that revolve around project increments comprising of short Timeboxes (e.g., two to four weeks). The underlying principle throughout is that the level of detail of planning is inversely correlated to the length of the time horizon driven by the level of uncertainty over time. Planning must accordingly be iterative taking on greater detail as plans evolve. For example, Pre-Programme and Programme Feasibility planning is restricted to the selection of undertakings and their cost-benefit assessment in terms of deliverables and risk. It is not until the Programme Foundations that definition and prioritisation of high-level capabilities (mapped to defined benefits) occurs along with clarification of the route towards realisation of the Business Architecture Model. This situation is depicted in Fig. 5.8 which shows the key elements that determine the evolving Programme Plan.

Detailed planning also occurs at the level of capabilities enablement since each capability is related to one or more projects during which reviews take place to examine how the tranche has performed (see Fig. 5.9). It is worth reminding oneself that since projects incrementally delivering their respective capabilities, benefits realisation must accompany capability enablement throughout the tranche (and not on its completion as might be the case in non-agile approaches). Benefits realisation itself requires planning since it is not only the initial roll-out and integration of new capabilities that is of interest but also the sustained and continual usage thereof.

The emphasis of planning and control at the programme level is therefore clearly focused on delivery of capabilities and the realisation of benefits in order to ensure project alignment with the programme whilst keeping control on time and cost bud-

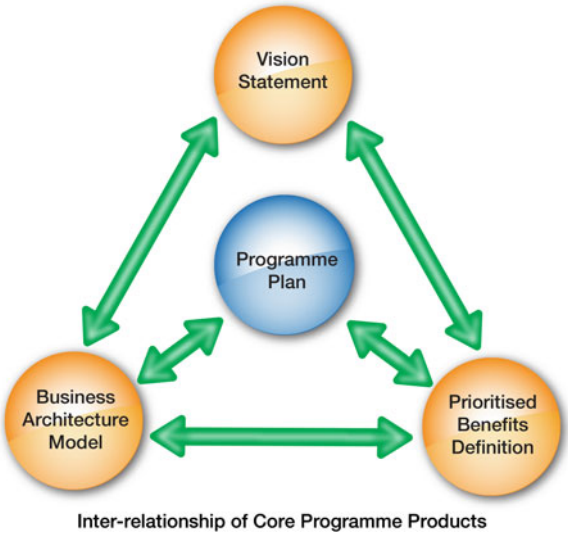

Fig. 5.8 Core Product Interrelationship © DSDM Consortium 2014. Reprinted with permission

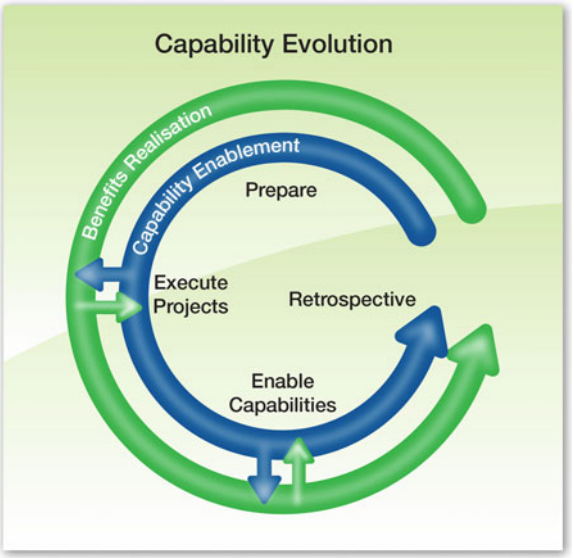

Fig. 5.9 Core Product Interrelationship © DSDM Consortium 2014. Reprinted with permission

gets and ensuring quality. This is achieved through incremental delivery and continual review as well as the learning that arises from consumption of budget (in relation to benefits realised) and the feedback loops between the capability delivery teams and their customers. The communication that is necessary is considerable at the programme level and requires that key roles engage at both the programme and project levels (e.g., attending project Daily Stand-Up meetings). When compared to other enterprise agile methodologies (e.g., SAFe®) it is less clear how programme and project teams ought to integrate. This notwithstanding, it may be fair to imagine that the Programme Technical Co-ordinator forms a virtual team with the respective Technical Co-ordinators of active projects or that the Programme Manager maintains a close relations with Project Managers at all times.

As illustrated in Fig. 5.10 there are several review gates (some of which imply the baselining of artefacts) at which various assessments are made. For example, prior to entry into Programme Feasibility the programme must be deemed to fit with the business strategy and on emergence from the Programme Feasibility, Programme Foundations and each Tranche Review the viability and continuation of the programme must be determined. At the tranche level more detailed reviews are conducted that take into account what capabilities have been delivered and whether or not benefits have been realised. The important point to bear in mind in such reviews is that they exist to implement governance thereby adding value to the programme. They ought not therefore represent costly and timely overheads but rather should facilitate programme level learning. The very fact that such activity is concentrated at the tranche level makes this possible since the short timescales and tight feedback loops support these goals.

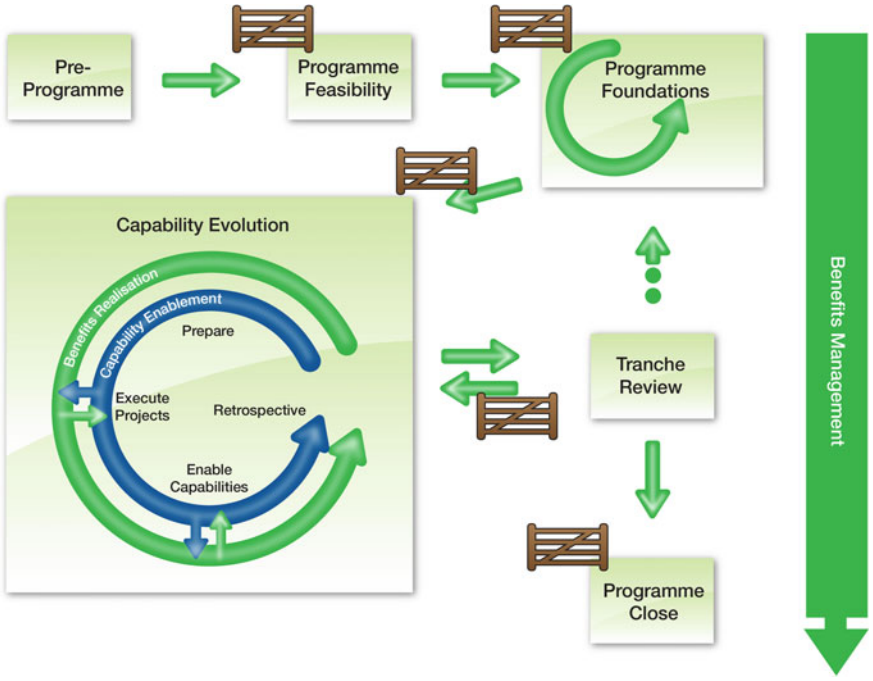

Fig. 5.10 Agile Programme Review Gates © DSDM Consortium 2014. Reprinted with permission

Finally, whilst the programme views increments at the tranche level, projects continually produce deliverables at much shorter timescales. If already using Timebox Review Records, then these can be picked up at the programme level without the need to burden individual projects with programme reporting requirements as described earlier in this chapter.

5.6 Management Implications

The discussion of agile programme management has taken place within the context of strategic portfolio management, a subset of which constitutes on-going active portfolio management in an environment of hybrid approaches and methodologies. Though appealing in itself, this rather top-down view is not shared by all agile practitioners as can be deduced from the emergence of lean startup philosophies. Thus agile programme management as presented here represents an evolution of existing views concerning the structuring of change activities that has been imbued with agile characteristics and practices. Agile programme management as proposed by DSDM therefore exhibits different characteristics to other enterprise agile methodologies (e.g., DAD, SAFe®). Whilst DAD appears to embody a very broad range

of practices all embedded into a decision-making framework with which just about any conceivable pair of techniques can be married together as circumstances dictate, SAFe® cites the use of Architecture Release Trains that mirror the cadence of individual teams back into the wider programme and provide business and technical support infrastructures to ensure that programme increments can be delivered. Part of the fluidity of SAFe® might be attributable to the fact that it designates agile, rather than project, teams at its lowest level and these may be involved in a variety of change and other related activities (e.g., Scrum or Kanban teams). This enables it to dovetail into the mainstream "Scrum of Scrums" thinking wherein higher levels of coordination are achieved by getting representation of individual agile teams (often, but not necessarily, their respective Scrum Masters) to convene on a regular basis to discuss coordination and integration issues. Complicating the situation for the DSDM Agile Programme Management Framework is its admission of both agile and non-agile approaches (which SAFE® does not accommodate) and this in part may account for the striking resemblance between it and traditional frameworks such as the Managing Successful Programmes®. Thus there is a desire to retain the traditional roles and structures and attempt to make them amenable to Agile (e.g., encouragement of empowerment, adoption of iterative and incremental approaches). Whilst non-agile participants may feel at home in such environments, they need still to understand that the conduct of daily business has in fact changed and, cultural differences aside, the conscious reinterpretation of otherwise familiar terms (e.g., Programme Manager, Tranche) must take place in an agile manner if the undertaking is to succeed. There is thus the risk that terminology and practices, wrapped in vaguely familiar language, will be take as a license to continue on as before. For example, the implicit hierarchy between the Programme Technical Architect and the project level Technical Co-ordinators must respect the autonomy and empowerment within the respective project teams and the tension that will inevitably exist in such arrangements need to be skilfully resolved.

Reporting at programme and project levels needs to be agreed early on in the programme and appropriate structures put into place. Close reading of the DSDM Agile Programme Management Framework indicates that insofar as the DSDM Agile Project Management approach is in place there ought to be no issues since much of the necessary project governance and review is already in place. Thus signalling up this information using RAG[5] charts may, for example, suffices for the most part. A greater challenge arises if other agile methodologies that have less of a focus on governance (e.g., Scrum, XP) or non-agile approaches (e.g., PRINCE2®) with their own cultural nuances are being integrated. In the case of the former the issue is not with the agile programme approach but rather the inherent weaknesses in the underlying project methodology (e.g., use of forced ranking rather than prioritised backlogs) which simplify reinforces the point that early discussion of the matter is called for.

[5] RAG is a common reporting acronym for Red, Amber and Green status reporting.

Chapter 6
Governance

Abstract Governance is concerned with how decision-making within programmes and their projects is conducted. Embedded in this view is the assurance that the necessary project management capability exists and is sufficient for the needs of the organisation and that at an individual project level resources are being deployed in an effective and efficient manner. In respect of organisation and role structures governance also encompasses not only matters of accountability and responsibility but also the concentration and balance of power. Thus whilst traditional governance models often stipulate the existence of a steering committee that directs projects and to whom decisions may be escalated for resolution, other approaches rely more on collective wisdom and the democratisation of decision-making suggesting that the implementations of governance models may vary considerably in their details. Therefore, although governance seeks to ensure that intended benefits are realised in a manner that ultimately fulfils the overall strategy and vision, precisely how this is achieved is an open matter. This implies on the one hand that accommodation of agile thinking is called for and on the other that agile stands to benefit from traditional perspectives on governance.

6.1 Introduction

Governance is concerned with the nature of decision-making frameworks that guide and direct activities and as such is a separate activity from the management of resources and people [11, 131]. Broadly speaking there are two main influences on the wider definition of project governance that are derived from transaction cost economics and principal-agency theory. Transaction cost economics examines the costs incurred when conducting a transaction (e.g., searching for information, contract bargaining and enforcement) and became a pivotal economic theory that gave rise to the notion of economic governance structures that shape economic activity and its control [278]. The prevalence of transaction cost economics is owing to the significance it plays in overall costs and therefore its relevance to governance concerns the effective and efficient use of resources. It should therefore come as no surprise that contractual machinations (wherein the interests of different parties need to be

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_6

protected and monitored in order to ensure that key participants do not attempt to manipulate outcomes solely to their benefit) are as much a governance concern as the balancing of the needs of competing and conflicting interests.

Despite broad connotations of decision-making frameworks, specific definitions of project governance [4] range from those focusing on economic contractual details [261] to those that adopt a very broad principled stance on stakeholder management, procedural justice and contractual obligations [219]. In some cases project governance is understood as the projection of corporate governance onto the project in order that its activities are aligned with the objectives of the organisation [11]. However, more often than not the characteristics of governance are conditioned by many factors including the asset specificity of project deliverables, the relational nature of the undertaking and the prevailing business culture. These considerations admit the possibility that governance might also need to be tailored in order to cater for the management of relationships amongst the parties involved [13]. It only follows, therefore, that adoption of Agile will inevitably also have some bearing on how governance will be expressed within a project.

In spite of the diversity of debate concerning the nature of project governance there is a clear dichotomy in terms of how it comes to influence individual projects. On the one hand there is external project governance that is imposed in a top-down fashion onto projects. Here the objective is to establish independent policies and standards which projects must follow and on the basis of which comparisons across projects can be made (e.g., governance function found at the portfolio or programme level and perhaps executed by a PMO). This perspective highlights principal-agency concerns between individual projects and the overriding authority that dictates the terms of governance to them. This points towards a driving need for alignment with an overarching strategy (e.g., as in the situation where projects make up a programme) as well as the necessity to subjugate projects to standardised monitoring and reporting. The other, moderately more popular view, is that project governance is subject to the specific circumstances of a project and therefore entails tailoring and establishment of shared rules and understandings between participants and their stakeholders. Indeed, governance in this context is understood as a means of facilitating goal congruence of parties that hold differing views [279]. Thus the key features of this perspective focus on the transitory though independent nature of project organisations bound together by a shared goal, the often complex relationships between parties (e.g., suppliers, contractors) and the role of governance in safeguarding the alignment of interests and accommodation of external contingencies.

In common with most views on governance, decision-making ought to involve appropriate individuals based on their decision authority (e.g., command of resources) and the ability to make correct decisions in a timely manner (e.g., proximity to relevant operational details). In light of the systems dynamics observations of Chap. 1, timeliness is particularly important since the very structures and interfaces that are often introduced in the name of good governance can in fact impair decision-making by introducing delays that result in decisions being made on the basis of stale information. Thus frequent review points throughout the process are required in order to continually gauge the suitability of the emergent solution and to adapt its development

in light of new information (e.g., changes in the market place). There are seemingly countless principles surrounding project governance though the following, that have been adapted and reinterpreted in agile terms, enjoy a reasonable consensus [86] amongst project governors:

- *Single point of accountability*. The responsibility for any undertaking must be traceable back to a single credible individual who is in command of the necessary resources and authority to direct activities and assume accountability thereof. Inherently linked with leadership, this individual must be have the capacity to act as final arbitrator of conflicts concerning the direction and nature of the undertaking. In DSDM this role is assumed by the Business Sponsor at the project level and the Business Programme Owner at the programme level. These DSDM role assignments are based on the DSDM Agile Project Management and Agile Programme Management Frameworks respectively. Other agile methodologies also accommodate this principle as evidenced, for example, by the Product Owner role in Scrum. Though incorporated in a single individual, such a role is expected in an agile environment to solicit feedback and involvement from others.
- *Ensure alignment of project governance and solution delivery*. This principle advocates the direct involvement and integration of Business at multiple levels into decision-making concerning solution development (e.g., Business Ambassador, Analyst and Advisor roles in DSDM). It also ensures that divisions (perhaps arising from organisational structures) do not inhibit communication and collaboration and that there is continual validation in relation to what is being delivered.
- *Separation of stakeholder management and project decision-making*. Whilst stakeholder management is important and ensures acceptance of solution deliverables it is not a mandate to involve all stakeholders in all decisions relating to the project. The wider disagreements and conflicts over the direction of a project that invariably arise amongst different stakeholders should not be allowed to undermine either the authority of the project sponsor or inhibit the empowerment and self-organisation of the Solution Development Team (e.g., by perhaps involving the right stakeholders at the right point in time). Adoption of this principle clearly favours less democratic governance models but is nonetheless consistent with agile practices insofar as the previous governance principle is honoured.
- *Separation of project and organisational governance*. This advocates the independence of project decision-making from organisational influence (e.g., individual line managers) though it does not preclude the projection of corporate governance onto projects. In agile environments this is particularly important since much decision-making takes place in the context of self-organised teams that must be sufficiently empowered.

One aspect of governance that warrants special attention for agile undertakings is the manner in which decisions are affected by social context [95] suggesting that the social networks in which participants operate play a role in how decisions are arrived at. For example, shared mutual history and trust may influence how working agreements are structured. A working agreement in this context is an consensual statement concerning how a team intends to work together that highlights shared

values and desired behaviours (e.g., timeliness at meetings, willingness to discuss points of conflict constructively and openly). Clearly, given the socio-technical characteristics of Agile together with its empowered self-organisational structures, this is an important detail to consider when assessing governance in agile environments. Thus monitoring of progress towards delivery of benefits in agile environments tends to focus on the extent of the viability of social contracts, the achievement of functional outcomes based on consensual decision-making together with the effectiveness of the learning processes that alleviate the initial uncertainties or cope with marketplace vagaries. The abilities of adaptive devices (e.g., Prioritised Requirements List, MoSCoW prioritisation) and the effectiveness of self-organisation (incl. decision-making and conflict resolution) to consistently deliver that which is deemed most valuable at any given point in time thus take center stage. Accordingly, of governance interest are those parameters (e.g., methodological configuration, team composition, disposition and stability) that best contribute towards focused delivery of value in spite of the change and uncertainty that may surround them. Therefore Agile neither argues that governance is no longer necessary nor that traditional governance bodies have no further part to play, but rather, as is so often the case, a different and more nuanced approach is required wherein the configuration of organisation and people is measured in terms of its capacity to deliver value in turbulent environments.

In many respects, Agile enhances traditional aspects of governance through its increased speed of communication which results in enhanced timeliness of information on which decisions are based. Furthermore the superior validation that arises from integrated heterogeneous teams also plays a key role. Moreover, at least in the IT sector, there does appear to be a tentative positive correlation between the practice of the principles of the agile manifesto and commonly held success factors in the adoption of IT governance [155]. For example, the prioritisation of early and continuous delivery of valuable solutions along with the integration of business and technical into a single team both support the oft cited success factor that the involvement of senior management (e.g., Business Sponsor) is essential for sponsorship, prioritisation and decision-making.

6.2 DSDM and Governance

Since its inception DSDM has transferred several traditional project governance principles and practices and embedded them at multiple levels into its own process model (e.g., review points throughout the entire lifecycle) and procedures (e.g., production of Timebox Review Records). DSDM envisages that the burden of governance is borne in large part by the project level roles within a project (e.g., Business Sponsor, Business Visionary, Technical Co-ordinator, Project Manager) in a manner that reflects the traditional steering committee structure found in traditional programme and project management though it cautions against an overt control mentality. Generally speaking such a steering committee, to whom the Project Manager, Technical Co-ordinator and Business Visionary would be expected to report, resides outside

of the project and ought to include the Business Sponsor alongside other relevant senior parties. In practice, DSDM is considered highly adaptable in respect of organisational constraints and is capable of accommodating most of the common traditional governance structures in terms of the following activities:

- *Business Focus*. The requirement that the project purpose and objectives be outlined in a Business Case and that the necessary business, solution and management foundations be developed in the early phases of a project (i.e., Pre-Project, Feasibility and Foundations) communicate clearly the intent of a project enabling it to be reviewed for fitness of purpose.
- *Structured Approach*. Since a project is a temporary change vehicle entrusted with resources in order to achieve specific goals, it ought to have a structured approach from which it is possible to deduce whether or not the undertaking is a going concern. This is evident from the phased lifecycle model of DSDM which clearly delineates the main activities (e.g., initiation, development, deployment).
- *Roles*. Clearly defined roles and responsibilities (see the Appendix C) ensure representation of business and technical views and identify project approach risks that might call into question the validity or integrity of the project. This includes having a single person accountable for the existence and performance of a project (i.e., Business Sponsor) consistent with the governance principles outlined earlier. In addition some of the project level roles (e.g., Business Sponsor, Business Visionary, Technical Co-ordinator) may constitute a project board to which the Project Manager must report should this governance model be desired. Further details can be found in Chap. 4.
- *Stakeholder Involvement*. The "one team" integrated approach provides for inclusion of business and technical representatives at many levels (e.g., planning, review, testing and solution demonstration) thereby ensuring meaningful and actionable involvement of the key players.
- *Delegated Decision-Making*. Empowerment within DSDM projects is discussed during the Foundations phase in order to determine what scope team members have in decision-making. For example, MoSCoW prioritisation provides the basis for determining the scope of autonomy of the Solution Development Team and the contingency of a Timebox.[1]
- *Review Points*. Reviews offer the opportunity to assess progress towards project goals and therefore constitute an important element of project governance. These include the reviews that take place at the end of Investigation, Refinement and Consolidation Timebox subphases and the retrospective activities that occur at the end of each Timebox.
- *Auditable Decision-Making*. As necessary, decisions and feedback of relevance to the project (e.g., amendments to acceptance criteria) are expected to be recorded (e.g., Timebox Review Records) making it possible later to understand when and

[1] A Solution Development Team may refer issues concerning Must items on the prioritised Requirements List to the Business Visionary or Sponsor but be empowered to make their own decisions regarding Shoulds and Could items.

how they were arrived at. More significant matters (e.g., change of scope, project constraints or the business case) must be decided on by the project level roles and escaled, where appropriate, to the Business Sponsor.

- *Transparency*. Accessibility to the evolving solution is the clearest indication of progress on a project. Furthermore the openness generally found in agile environments with respect to reporting and tracking also provides oversight and assurance of progress towards the project goals. This transparency is necessary in order to exploit the feedback mechanisms employed by agile projects.

As indicated in Fig. 6.1, governance is also manifest in the artefacts produced by DSDM ranging from initial scoping documentation (e.g., Feasibility Assessment, Foundations Summary), continual review processes (e.g., Timebox Review Records) through to retrospective assessment of accrued benefits (e.g., Benefits Assessment, Project Review Report).

DSDM, rather like other enterprise agile methodologies such as SAFe® and DAD, goes some way to countering claims that governance in Agile is weakened by an overt focus on product development or engineering techniques and practices. Indeed, the similarity with traditional governance structures becomes more apparent at the programme level where a clear influence of external project governance emerges though this is qualified by the advice to push decisions down to the lowest possible level. This resemblance is perhaps a consequence of the intention of the DSDM Agile Programme Management Framework to accommodate both agile and non-agile project approaches into a single agile programme. Three levels of governance at the programme level are envisaged and these lie under the control of the Business Strategy Team, Programme Board and Capability Teams respectively. Each level interacts with the Portfolio Management Office, Programme Support Office and the Project Support

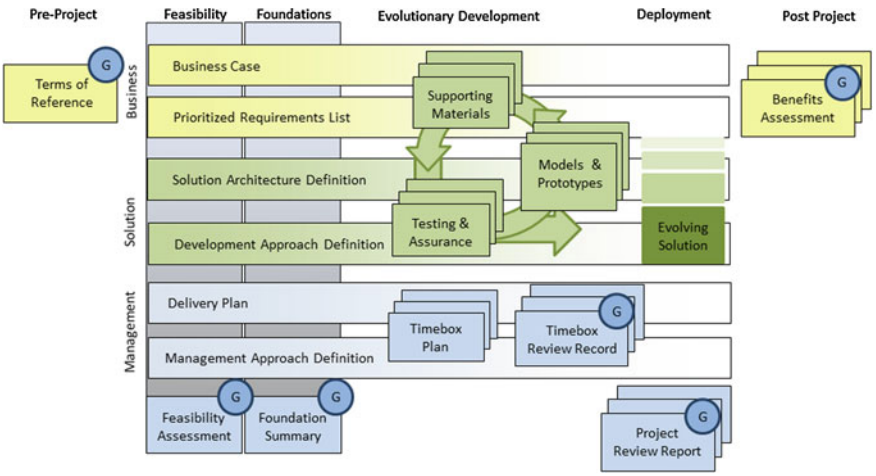

Fig. 6.1 Role of Governance in DSDM Documents. Published with kind permission of © Alan Moran 2015

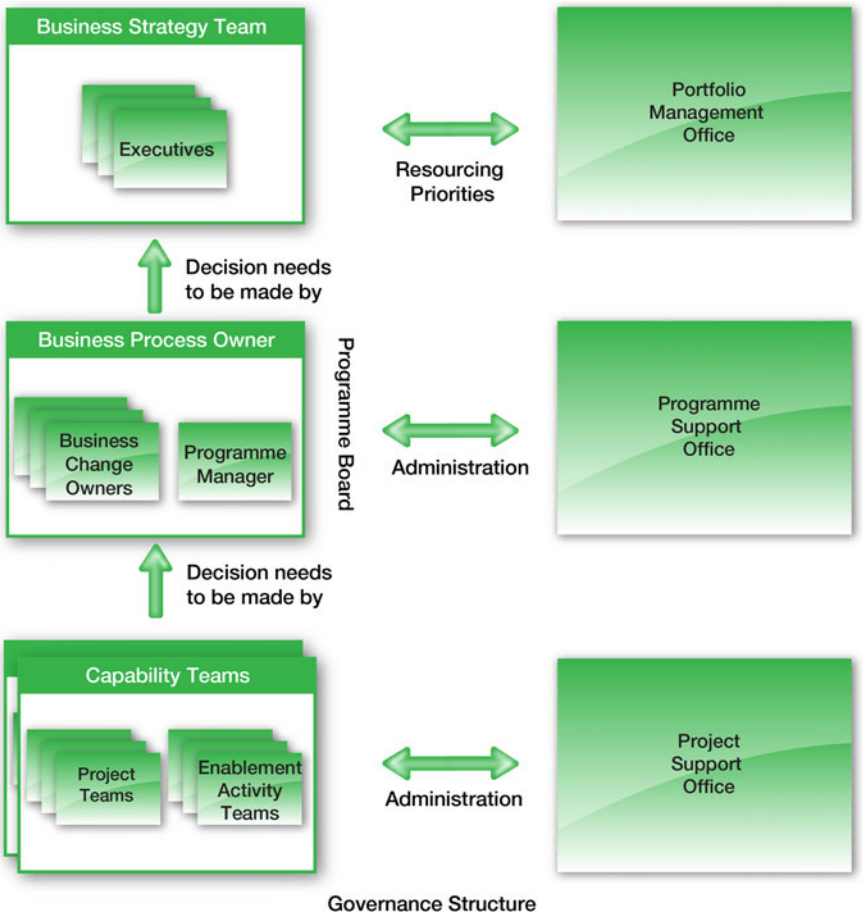

Fig. 6.2 Agile Programme Management Governance Structure © DSDM consortium 2014. Reprinted with permission

Office as indicated in Fig. 6.2. Together they facilitate the management of resources (e.g., in situations of scarcity of resources or conflicting business priorities) whilst establishing an escalate up, delegate down hierarchy of decision-making between the levels.

The composition of the Business Strategy Team is usually senior executives who have command over resources and funding and who direct corporate strategy. Assisting in this respect is the Portfolio Management Office who help in the prioritisation of programmes and projects in line with business needs and priorities. The Programme Board must also focus on programme consistency and high-level tranche and project planning as well as the management of necessary resources. The Programme Board may be supported by the Programme Support Office in planning, reporting and administration functions. Finally the capability teams will be subject to

the project level methodology and practices (e.g., DSDM Agile Project Framework) with the backing of the Project Support Office that supports them in administrative tasks. The review points at the project level discussed earlier are augmented with tranche review points at the programme level.

Needless to say this is a considerable overhead that risks becoming a burgeoning bureaucracy unless conscious efforts are made to keep it lean. A documented Governance Strategy is intended to communicate intentions during programme setup in order to help stakeholders navigate decision-making through these structures and to ensure the protection of autonomy at lower levels. In order for delegated authority to function, some thought needs to be given as to what decisions affect which levels. For example, programme level decision-making must necessarily cover overall vision, budget and high-level project scope but could also include technological matters that impact project interfaces (e.g., where neither affected project may be willing to enter into the fray). Many remaining decisions can therefore safely be pushed down entirely to the project level. Most cases do appear, however, in practice to be relatively clear with the majority of issues arising in situations where projects have either scope overlap, dependencies or interfaces suggesting that most governance issues ought to be able to find their focus appropriately.

6.3 Management Implications

Projects typically arise amidst circumstances of incomplete information wherein planned activities must chart a course towards a desired goal or outcome that involves minimal amount of risk (incl. the exposure of stakeholders to moral hazard). Within this context it is the conditions surrounding the project (e.g., highly innovative environment, strict regulatory controls) that often determine the key characteristics of the governance model to be employed. Thus, whilst it is tempting to reduce governance to a decision-making framework, the reality is that not all governance frameworks are alike and that there is considerable divergence of opinion on how governance ought to be implemented (e.g., SLAs, steering committees, democratisation of power). This notwithstanding, governance remains a reality for agile projects since it is legitimate to question whether or not resources are being effectively and efficiently deployed and if the basis on which decisions are being made is sound.

In practice delegated decision-making means that decisions are pushed down where possible and that teams are empowered to make decisions themselves within an agreed scope (e.g., based on MoSCoW levels). An often overlooked detail is that they are also *required* to make such decisions and must themselves assume this responsibility rather than push uncomfortable or difficult decisions up the chain of command. In most cases support functions will not assist in this regard since both the Programme Support Office and the Project Support Office exist to provide administrative rather than managerial support. This point ought not to be underestimated in organisations that are transitioning to Agile or in programmes where there is a mix of agile and non-agile approaches. Despite the optimistic wording of the

DSDM Agile Programme Management Framework, cultural issues and notions of self-responsibility do repeatedly come to the fore and need to be actively managed. Nevertheless the various DSDM frameworks do contribute postively to the overall governance debate and counter the criticism of Agile that it is weak on governance.

Existing governance models based on traditional frameworks (e.g., MSP®, ITIL®, COBIT®) may also continue to serve a useful role in agile environments though perhaps not without some modification since embedded within them are assumptions of command-and-control and separation of duties that are not entirely beneficial from the agile perspective [131, 197, 198]. For example, organisational impediments arise when functions are segmented in such a manner that activities cannot be performed by a single team. An example of this is the COBIT®PO4.11 "Segregation of Duties" that is justified on efficiency grounds and is claimed to reduce risk in relation to damage and compliance. As evidenced by the programme and project governance structures of DSDM, however, it does seem reasonable to expect that agile approaches can co-exist peacefully with traditional ones provided that mutual accommodation is assured. Furthermore, as discussed in Chap. 9, the use of configuration management as a tool in the governance arsenal fits well in agile environment that adopt data-driven management practices. This also ensure the baselining of key artefacts and tracking of change that does not impede the flow of the team.[2]

Increasingly governance relies on evidence-based decision-making processes that in turn leverages the high automation levels often found in agile environments. This provides one source of input into an independent assessment of process and solution quality that can instill continual improvement practices within the team. Such metric based approaches ought, however, to be determined by the entire Solution Development Team rather than handed down from above in order to satisfy reporting requirements. Employed solely as a basis for promoting organisational learning, such metrics do seem to acquire some immunity towards the manipulation or massage that might otherwise arise (e.g., if they are used to assess performance as part of bonus schemes). Such matters generally need to be taken into account by a governance board when determining what criteria warrant intervention in a project so that these do not corrupt the overall process.

[2] Rather surprisingly configuration management and the baselining of key documents does not feature much in other agile methodologies (e.g., Scrum, XP) though it is obliquely alluded to in some (e.g., DAD, SAFe®).

Chapter 7
Quality and Test Management

Abstract Classical approaches to solution development envisage quality and testing in terms of completed solutions that are to be assessed for suitability resulting in either transition to production or return for rework and amendments. Such processes possess gatekeeper characteristics and use existing specifications as their grounds for assessment. In an agile environment, quality and testing are, however, more deeply integrated into the solution development process and assume the nature of dialogue within the team concerning the meaning and implications of quality and how it is to be concluded that it has been achieved. The agile toolkit includes many practices that support this dialogue including direct collaboration and communication, pair programming and code reviews and technical driven practices such as refactoring, test-driven development and continuous integration. This enables business to clarify its position on quality, affording the team the opportunity to have this validated (e.g., posing questions to Business Analysts or performing reviews at the end of Timeboxes) and to continually improve its own process (e.g., conducting Timebox retrospectives, use of automated quality and testing tools). The result is that quality and testing activities occur earlier and more frequently than in non-agile approaches and that in spite of the occasional difficulty of clearly delimiting quality and testing activities (e.g., quality control), there can be no doubt of their presence or efficacy in agile environments.

7.1 Introduction

Quality and testing are notions that are inherently linked though they in fact represent very different perspectives on the solution development process. In spite of some very distinguishing quality and test characteristics of Agile (e.g., process reflectivity, test-driven development) some aspects remain largely unchanged such as the traditional distinction between static (e.g., review) and dynamic (e.g., testing) techniques [73], though there is a considerable emphasis on automation (e.g., continuous integration, static code analysis) which contributes to visibility and feedback. For example, a solution team may define early on in the project a specific test coverage target along

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_7

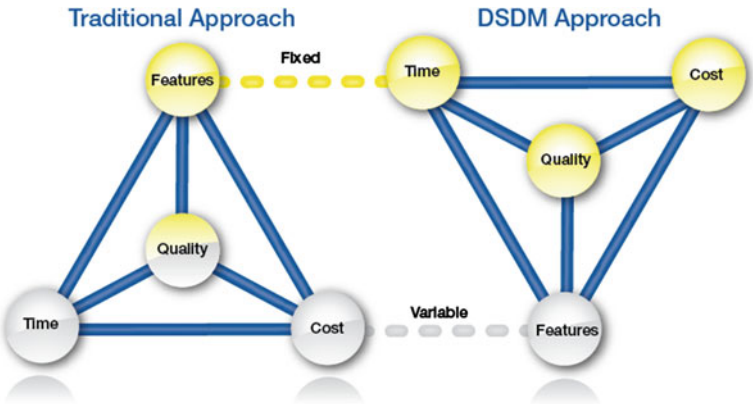

Fig. 7.1 Comparison of Traditional and DSDM Project Variables © DSDM Consortium 2013. Reprinted with permission

with desired technical quality attributes (e.g., package tangle index targets[1]). Progress towards these targets may be displayed on an information radiator providing continual actionable feedback (e.g., specific parts of the solution that fail to satisfy quality requirements).

Traditional definitions of quality and testing have tended to exhibit a rather strong contextual bias towards manufacturing and product development with emphasis being placed on satisfaction of needs through product features and freedom from defects [140], uncompromising conformance to requirements [57] as well as the balance of cost and benefit as perceived by the user [267]. When transferred over to IT, quality has come to be understood in terms of functional correctness, integrity of design (e.g., reusability, extendibility) and sustainability (e.g., maintainability, portability) as described in [166, 172, 248]. As a result the defence of agile quality has revolved around the recognition of how its practices contribute to these elements. For example, correctness and ease-of-use are said to be attributable to the presence of business representatives (e.g., Business Ambassador and Business Analyst) in the solution team whereas timeliness, maintainability and cost-effectiveness are derived from interative development specific methodological principles (e.g., on-time delivery and iterative development as described in Chap. 1).

As illustrated in Fig. 7.1, DSDM considers quality, like time and cost, to be fixed project attributes, leaving scope as the only real variable. This is in keeping with the view of the wider agile community though DSDM is rather more explicit on the need to fix quality prior to start of solution development and to review it continually throughout the project.

[1] Package tangle index measures cyclical dependencies between individual units and packaged components of software. High values suggest that code will be more difficult to maintain, that changes to the software may result in unexpected side effects and that a higher cost of testing will result.

It would seem that there is also a natural alignment of agile practices with quality and testing thinking [10]. For example, rather than treating acceptance criteria as documented artefacts, it is common in agile environments to formulate a suite of tests that actively verifies fulfilment of a requirement. These are retained within the solution as a means of tracking progress towards the requirement[2] and detecting changes that create inconsistencies amongst the requirements.[3] Since agile team members become the custodians of their own quality, collectivism endears higher degrees of commitment towards quality, leading to less need for formal quality assurance measures. Indeed, there is a directness and immediacy of agile practice which together with group transparency and individual generalist skills, enables establishment and maintenance of a high level of quality that is validated throughout by testing and review.

7.2 Quality Deconstructed

There are five major approaches to defining quality, each with their own particular emphasis and focus, and these have led to divergent views as to what precisely constitutes quality in a given context. Indeed, within organisations it is a common mistake to attempt to settle on a single definition of quality for the entire solution development process. Instead, each of the major five perspectives offers important insights concerning different aspects of quality (e.g., user expectations, solution performance and features) suggesting that specific definitions should be applied to different phases of solution development, in order to address the needs of different stakeholders. The following summary presents the key characteristics of each of the main approaches to quality:

- *Transcendent*. This approach defines quality as something inherent in a product or service reflecting universally recognisable high standards and achievement. This notion of quality is often linked with aesthetics and beauty and is something that must be experienced directly. This approach can, however, be problematic in terms of consistency of assessment.
- *Product-based*. Product quality relies on adherence to precise and measurable criteria that enable the ranking of products and services on the basis of a quality attribute (e.g., the extent of the presence or absence of an attribute). The ISO 9000 definition of quality as the "degree to which a set of inherent characteristics fulfils requirement" [133, Sect. 3.1.1], is consistent with this perspective, since it refers to permanent characteristics rather than that which can be assigned by association (e.g., the social status linked with using a product). Pursuit of quality according to this definition is often linked with higher costs.

[2] Tests that have been created without a corresponding solution implementation are considered failed and thus offer evidence of an incomplete solution.

[3] Often changes required to implement a requirement may clash with other requirements resulting in hard to detect side effects. Test-driven development and continuous integration practices make such conflicts relatively easy to uncover.

- *User-based*. This perspective defines quality as that which best satisfies the needs or wants of the user. In this context, quality is very subjective and may be linked to judgement of fitness for purpose (i.e., the extent of actual utility of the solution) and fitness for use (i.e., the extent to which a solution is judged to be capable of fulfiling its purpose). Despite this customer value based focus, preferences may yet exert an overriding influence on quality assessment.
- *Manufacturing-based*. Conformance to requirements is considered here to be the key quality criterion in this approach which judges products or services that deviate from specifications to be of inferior quality. Intolerance of defects, strong overtures of process control and an eagerness to get a solution right the first time are common hallmarks of this definition.
- *Value-based*. This econometric perspective frames quality in terms of performance at an acceptable price or conformance at an acceptable cost. The potential conflation of quality and value can make this definition difficult to apply in practice.

In relation to manufacturing-based definitions, a word of caution is warranted concerning the use of the term conformance as it is often used interchangeably with the term compliance, both of which relate to the relationship between the requirements and their implementation in a solution. A solution is said to be conformant if all the requirements are implemented, though other implemented elements may exist that bear no relation to the requirements. On the other hand a solution is compliant if everything it implements is in accordance with what was expected, though it is possible that there remain required elements that are yet to be implemented [257]. Either term can be appropriately qualified based on circumstances (e.g., partially compliant, fully conformant). Figure 7.2 illustrates the difference between compliance and conformance.

Owing to their repeated occurrence in the above definitions, a deeper understanding of the definition of quality can be derived from an analysis of the eight dimensions of quality [88, 89] listed below. These dimensions not only offer explanatory value, but also help to decompose some of the approaches to defining quality into their constituent parts as illustrated in Fig. 7.3. This latter point becomes important later when the DSDM definition of quality is reconstructed in terms of these dimensions,

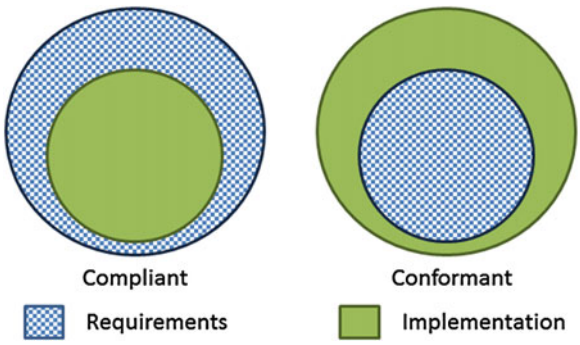

Fig. 7.2 Compliance versus Conformance. Published with kind permission of © Alan Moran 2015

Fig. 7.3 Mapping of the Major Quality Definitions in terms of the Eight Quality Dimensions. Published with kind permission of © Alan Moran 2015

in order to uncover how it aligns with the traditional approaches to defining quality. The eight dimensions of quality are:

- *Performance*. This describes the primary operating characteristics of a solution and is of particular relevance to product-based approaches to defining quality.
- *Features*. This dimension concerns elements that supplement performance characteristics [141]. Like performance, it is a common element found in product-based definitions of quality.
- *Durability*. This dimension is a measure of operating life in technical and economic terms. This is influenced by the ability to maintain a solution in a cost efficient manner which in turn is impacted by reliability standards. Accordingly, it is a desirable element of product-based quality definitions.
- *Reliability*. The likelihood of operation without failure is more of relevance to durable goods than to products or services. It is therefore a key dimension for manufacturing based approaches. In the software domain, this dimension may be understood as a non-functional aspect that affects solution warranty.
- *Conformance*. This reflects the ability to match existing standards either internal or external and, like reliability, is commonplace in manufacturing based definitions.
- *Serviceability*. This dimension encompasses objective and subjective aspects of maintenance and considers timeliness and competence of repair related activities. It tends to apply more to the service components of a solution (e.g., handling of incidents by a service desk).
- *Aesthetics*. Perhaps the most subjective dimension, this is focused on the appearance and impression of a solution (e.g., usability) and is deeply linked to individual preferences. It features rather prominently in user-based definitions of quality.

- *Perceived Quality*. This describes the impact of indirect measures of quality and is rather susceptible to brand and image considerations. Like aesthetics, this too can be a highly subjective and shifting element of quality and is of importance in user-based definitions of quality.

Quality is intrinsically linked to value, though these two concepts are in fact distinct (see Chap. 1 for a discussion of value). Accordingly, it is to be expected that a focus on business value ought to be reflected in the definition of the quality of a solution. This, however, does not preclude the fact that quality may embody other nuances not intrinsically found in a purely utilitarian driven assessment of a solution (e.g., exclusivity, social status) that also have a bearing on the acceptance of a solution.

In light of the observation that an organisation must adopt multiple perspectives on quality, the question arises as to which definitions are most appropriate in general for the agile context. The capacity of agile processes to cope with change and risk through iterative practices suggests that the manufacturing-based approach is generally less suited since this definition may require up front specifications against which to assess conformity. This is confirmed by the fact that agile processes typically adopt an incremental compliance based approach to quality (i.e., building up an emergent solution) rather than an attitude that relies heavily on the reliability and conformance quality dimensions. Similarly the transcendent definition of quality is problematic to apply in the agile context, since it requires the application of aesthetic and experiential criteria to an evolving solution in which emergent characteristics arise gradually over time.[4] Whilst this in itself does not preclude agile transcendent quality, it does make it a difficult to adopt a practical stance on the matter using this approach. User-based definitions of quality are appropriate for the identification of those characteristics that will later ensure acceptance of the delivered solution and thus provide a legitimate basis for determining quality criteria (e.g., usability). Similarly, since feature and performance criteria need to be transformed into inherent attributes of solution design, a product-based approach to quality definition would also seem appropriate. Finally as value-based considerations are likely to feature to some extent in any quality management assessment that is driven by costs and pricing it does not seem unreasonable to expect this approach to influence perceptions of quality. Without delving further into project circumstances, this is about as much as can be said in general terms concerning agile definitions of quality.

Turning to the specifics of DSDM, the compliance based tone of its approach to quality management (e.g., requirements that are gradually met through fulfilment of the Minimum Useable Subset) along with a focus on fitness for purpose[5] based around user defined acceptance criteria, suggest that quality is rooted in user and product based approaches. This is clear from the DSDM Agile Project Framework literature which adopts a clear product-based stance [73, 11.4] as well as a strong

[4] Although solution developers often describe their outputs in transcendent language (e.g., elegant architectures) this usually refers to internals rather than any externally perceived quality attributes.

[5] Though the emphasis of reviews appears in the DSDM literature to be on fitness for purpose (rather than fitness for use), application of the principle of focus on business needs lends credence to the assumption that both are in fact implied.

affirmation of the need to focus on business value. This is further supported by the fact that the framework has been recognised as an appropriate approach for ISO 9001 accreditation and has been shown to have an appeal to organisations that wish to retain agility in CMMI environments of level three or higher. The underlying DSDM methodology urges a non-compromising stance on quality and employs a variety of techniques (e.g., facilitated workshops, review sessions, MoSCoW prioritisation and defined acceptance criteria) to engage stakeholders, clarify and align requirements with business needs and validate the evolving solution. Quality expectations are expected to be set during the early phases of a project (e.g., Foundations) and reaffirmed throughout (e.g., Timebox acceptance criteria). In addition, DSDM Agile Project Framework also defines the following three maintainability levels:

- *Level 1*: *Maintainability is a required attribute of the initial delivered solution*. This level prioritises a robust implementation of a solution that can be supported from the outset.
- *Level 2*: *Deliver first, re-engineer later*. This level favours prompt delivery of a solution accepting that this may entail reworking at a later date (referred to a technical debt) for which budget ought to be set aside. This approach can be justified if time to market is critical.
- *Level 3*: *Short-term, tactical solution*. Maintainability is not a concern as the solution is intended to be temporary. Clearly, a maintainable replacement needs to be planned and measures put into place to ensure that the temporary solution does not become permanent.

These maintenance levels are of interest since they reflect the serviceability, reliability and durability quality dimensions. For example, products developed as

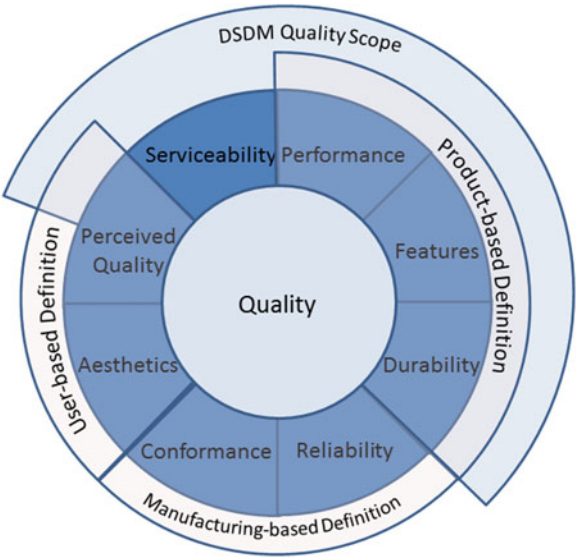

Fig. 7.4 DSDM Agile Project Framework Quality in Terms of the Eight Dimensions. Published with kind permission of © Alan Moran 2015

short-term tactical solutions are by definition not intended to be durable solutions and may exhibit poor serviceability. Whilst there is clearly a relationship between durability and reliability, the focus of the DSDM Agile Project Framework does appear to lean more towards the former rather than the latter, notwithstanding its justification in terms of operating costs, reliability and risk. Thus solution quality is framed primarily in product-based terms with some attention paid to user-based approaches along with some accommodation of serviceability and to an extent reliability as illustrated in Fig. 7.4.

7.3 Quality Control and Assurance

Any discussion of quality control and assurance requires first a clear understanding of the notions of validation and verification that underpin them. Validation refers to the assurance that a solution satisfies the needs of its users and other stakeholders. Verification is concerned with whether or not a solution is functioning in a manner consistent with specifications or requirements. In other words, validation is about whether or not the right thing is being done, whereas verification is about whether the thing is being done right.

Quality management traditionally distinguishes between quality control and quality assurance. Both terms grew out of manufacturing attempts over the period from 1930 to 1970 to model defects statistically and to apply appropriate quality and process control measures. These practices culminated in the 1980s in Total Quality Management (TQM) which was later superseded by Six Sigma. Quality control is a validation activity concerned with procedures and practices that ensure that a solution satisfies predefined quality criteria. It often operates on the principle that quality is determined in terms of metrics and can be verified through the testing of outputs (e.g., identification of defects, execution of software tests) though it is not entirely restricted to such activities (e.g., reviews as a quality control activity that does not include testing). It is thus an inherently reactive activity that endeavours to prevent defective goods or services arriving at the customer (e.g., by correcting any defects or removing the item entirely from circulation). Quality assurance is a closely related concept that is concerned with verification activities that focus on ensuring the quality of the process by which goods or services are produced. It is therefore a proactive undertaking that seeks to ensure that the process itself does not give rise to defects. Agile encourages a collective attitude towards assuming responsibility for quality that attempts to first stabilize and then continually improve the underlying process. Examples of quality assurance measures might include ensuring that everyone on the team is familiar with a specific agile approach (e.g., training) or that the environment is conducive to solution development (e.g., audits).

DSDM distinguishes clearly between quality at the solution (i.e., quality control) and process (i.e., quality assurance) levels. The former is an expression of quality control that seeks to validate if business needs are being met and if required standards (including maintainability) have been met and requires explicit

involvement of business (e.g., Business Ambassador and Business Analyst). This is facilitated through the integration of business roles in the solution team, discussion of requirements in facilitated workshops, explorative activities (e.g., modelling, iterative development) and clarity about what is really important and how this relates to business needs (e.g., MoSCoW prioritisation). The latter is concerned with quality assurance and addresses matters of the appropriateness of the project approach and the nature of its governance and in effect verifies that the process is being performed correctly (e.g., as outlined in the Development Approach Definition). In this respect, the approach adopted by DSDM is entirely in keeping with the traditional definitions of validation and verification found in classical quality management and testing.

7.4 DSDM and Quality

The quality conversation in DSDM begins during Feasibility (marked as item one in Fig. 7.5) where the broad parameters of quality are defined. This is carried over into Foundations (marked as item two) where it supports requirements scoping and establishes shared understandings in both functional and non-functional terms (recall from Chap. 1 that these are key determinants of utilitarian value). During iterative development (marked as items three to five), the focus switches to determining any acceptance criteria that may need to be refined during the Investigation subphase of the respective Timebox and executing tests to verify quality and record anything of importance. Finally full end-to-end testing ahead of deployment (marked as item six) ought to span a range of testing and related activities (e.g., performing trial runs of processes) including gaining assurance that the deployment itself was performed successfully. This approach illustrates well the agile attitude towards quality in terms of the parameters to which emergent solutions are expected to converge.

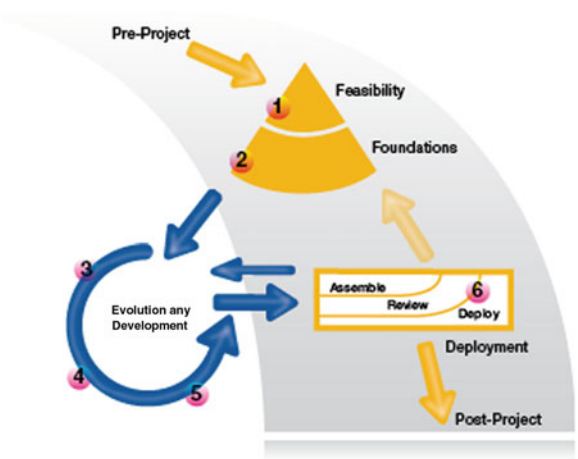

Fig. 7.5 Quality Control and Assurance in DSDM © DSDM Consortium 2014. Reprinted with permission

Ultimately, whilst it is the responsibility of key business stakeholders (e.g., Business Ambassador and Analyst) to clarify quality, it is down to the Technical Co-ordinator (under whom the traditional role of Quality Manager is subsumed) to ensure that it is upheld in the solution by ensuring adherence to standards and best practices. This includes promoting understanding of standards and practices (e.g., as outlined in the Development Approach Definition), establishing that the technical solution accommodates quality from a design perspective (i.e., Solution Architecture Definition) and ensuring that appropriate reviews are indeed taking place and that evidence of review activity is captured where necessary (i.e., the Timebox Review Records). Finally though the Solution Developer must implement quality (e.g., adherence to standards) it is the separate role of Solution Tester that embodies quality control. From the perspective of DSDM products (see Fig. 7.6) it is the Solution Architecture Definition that clarifies how solution quality is to be achieved and the Development Approach Definition that caters for process quality assurance. Thereafter, testing and assurance practices and the Timebox Review Records feature prominently in quality control.

At the programme level there is a conscious adoption of a "good enough" philosophy (i.e., the avoidance of gold-plating) though details are often left to individual projects unless they concern programme wide technical and organisational standards for which the Programme Technical Architect and the Programme Support Office respectively are responsible. Quality is therefore reduced to capability and business process quality (i.e., do the delivered capabilities support the realisation of benefits?) on the one hand and programme process quality (i.e., adherence to standards including ISO and CMMI and effectiveness of governance) on the other. In particular the incremental delivery of capabilities provides feedback points (e.g., pilot reviews) with which to assess whether or not benefits can indeed be realised as early as possible and whether sufficient capability has been delivered.

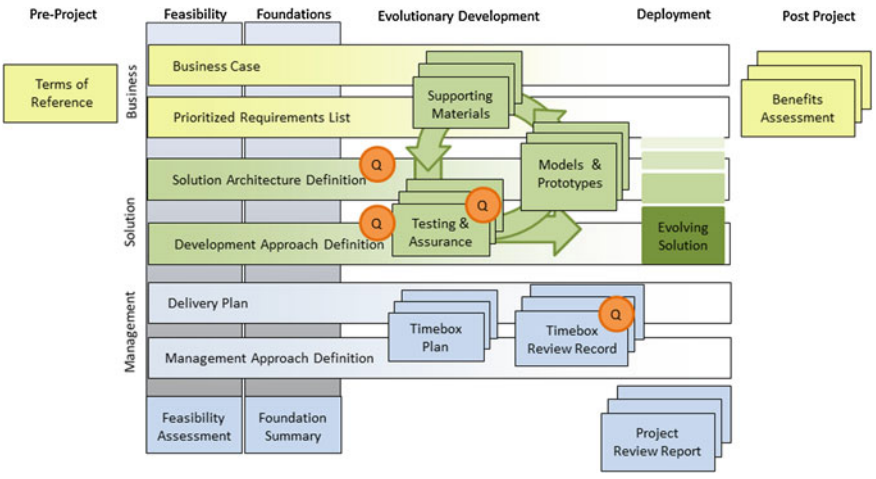

Fig. 7.6 Role of Quality in DSDM Documents. Published with kind permission of © Alan Moran 2015

7.5 Testing

Though distinct from quality, testing and quality are linked. Testing is, however, not limited to the mere validation of requirements and predefined quality attributes, but seeks also to trace the roots of defective behaviour in order to identify where appropriate preventative measures might be applied. Thus quality arises during design and is something that must be built into a solution rather than bolted on later, whereas testing is about establishing whether or not a solution exhibits desired characteristics and encompasses not only expected behaviours (positive tests), but also demonstrates that unexpected behaviours do not arise (negative tests). For example, it is as reasonable to expect that placing a car key in the ignition and turning it will start the engine, as it is unreasonable to discover that it also opens the back door.

To better understand this, it helps to distinguish between the terms failure, defect and error.[6] A failure is the inability of a solution to comply with a requirement or quality attribute (i.e., a deviation from specification or expectation). It is usually that which is most visible to the user and is often the focus of quality control exercises (e.g., solution inspection). A defect concerns a fault in the operation of a solution that leads to a failure and an error is that which led to the solution being in a state that gave rise to a defect in the first place.

> To understand the difference between a failure, defect and error consider a leaking radiator: the failure is a leakage that causes water to drip from the radiator onto the floor, the defect concerns the characteristics of the joint that was supposed to have kept the water in and the error is the design oversight of the engineer that caused that type of joint to be used in the radiator. This example illustrates how testing is ultimately related to quality through a chain of causality that links malfunctions with the processes that produced them. It thus becomes apparent that identification of a leakage and attributing it to a faulty joint is the role of testing, whereas designing a more appropriate joint and ensuring continual improvement of the design process are matters for quality management.

The primary role of testing is therefore to make defects visible. It is then down to solution development to resolve them so that failures do not arise and for quality management to tackle the underlying errors that give rise to the defects. What testing cannot do, however, is prove that there are no further defects in a solution. Furthermore, the sheer multitude of possible test scenarios makes exhaustive testing, impractical so any approach to testing must be capable of prioritising test cases. Indeed, experience suggests that solution testers need to be continually creative in their testing (e.g., by trying to find new ways to "break" the system) and be aware of

[6] The term "bug" is often used inconsistently in the IT sector. Not wishing to add to this confusion only the terminology failure, defect or error is used in this book.

the domain in which they conduct testing, allowing themselves to employ multiple prioritisation techniques (e.g., based on business value or risk) or blended strategies as outlined in [28]. Exploratory testing, defined as the simultaneous learning, test design and test execution (e.g., exploratory testing, sometimes considered a complementary approach to other agile practices, as described in [16]), may also be used to complement existing agile techniques (e.g., to diversify the existing test coverage or to explore areas that might require more systematic scripted testing).

The suggestion that implementation must proceed testing, thereby imposing a phased structure onto the solution development process, has been rigorously challenged in agile communities through a variety of effective testing strategies (e.g., test-driven development, continuous integration, regression testing). Thus testing ought to commence early and occur at different levels (e.g., unit, component, system and acceptance testing which employ a variety of black and white box strategies[7]) culminating in end-to-end testing conducted from the user perspective.

The eight testing principles of the DSDM Agile Project Framework share much in common with mainstream agile thinking regarding test management and capture the key features of modern testing. In spite of the obvious influences from the IT sector, their formulation makes them amenable to application in many other sectors.

- *Fail fast*. It is highly desirable for defects to become visible as early as possible. Numerous practices exist in support of this principle including early testing (e.g., use of quality control during development), regression testing (i.e., identification of appropriate tests as soon as defects are discovered and inclusion of these in regular testing) and integration testing (i.e., testing different components of a solution in unison).
- *Collaborative Testing*. This draws on the strengths of heterogeneity and diversity within the team by ensuring that test practices incorporate the views of different stakeholders.
- *Repeatable Tests*. This principle requires that appropriate test conditions are established in order that tests can be repeated (e.g., avoidance of side effects from previous executions of tests).
- *End-to-end Experience*. This principle advocates a holistic approach in testing that includes consideration of the elements usability, integration and performance that are of most significance to end users. Experience indicates that if this is not encouraged from an early stage then significant issues may arise at a later stage in a project.
- *Prioritised Testing*. This principle argues that tests ought to be prioritised in line with the requirements they validate (e.g., tests of "M" requirements should be given higher priority than those of "S" or "C" requirements). Accordingly this ensures good alignment of testing with value.

[7] Black box testing refers to the testing of a functioning of a product without knowledge of its internals (i.e., inputs are compared with expected outputs). White box testing, on the other hand, requires knowledge of how the product operates and needs access to its internals. Static code analysis and unit testing are examples of white box testing.

- *Independent Testing*. This can be considered an aspect of project governance insofar as a solution is to be tested by someone other than the solution developer who created it. It reinforces the "Collaborative Testing" principle and is the primary justification for the role of Solution Tester that is kept distinct from the Solution Developer.
- *Test-Driven Development*. This principle is about the determination of tests before the solution is created as a means of clarifying acceptance criteria. Tests for which no implementation are available are considered to have failed until a working implementation is provided.
- *Risk-Based Testing*. This is an alternative form of test prioritisation that emphasizes the need to ensure adequate levels of test coverage in areas of high risk.

Many of the testing principles are readily recognisable within the software development industry, which has embedded them for many years into its processes, albeit with an emphasis on unit and user acceptance testing at perhaps the cost of integration and system testing. For example, the practice of continuous integration wherein changes committed to a code repository triggers automated tests, is an excellent expression of the fail fast and repeatable testing principles.

Well-defined acceptance criteria provide a link between testing and quality management that leads to solution development with testability in mind. The test-driven development principle in particular encourages this mindset, since often deficiencies in solution design are often revealed by the interaction that occurs when they are tested. Owing to the fact that this is encountered prior to committing to the overall design, illustrates the positive quality influence that this principle has on solution design. Moreover, the integrated nature of the agile team lends itself favourably to collaborative and end-to-end testing, which underscores the importance of maintaining diversity within the solution team and ensuring the continued commitment of business. This collaborative spirit is reflected in the underlying business focus of DSDM which together encourages both prioritised and risk based testing principles.

The key DSDM Agile Project Framework roles involved in testing are the Business Analyst and the Solution Tester who play validation and verification roles respectively. The Solution Tester may also play an advisory role assisting in the writing of requirements or acceptance criteria though owing to the inclusive nature of agile teams, it ought not to be assumed that only the Solution Tester may create test cases (e.g., Solution Developers typically write their own unit tests). It is more likely, however, that a Solution Tester is more heavily involved in non-functional aspects of testing (e.g., performance tests) or ensuring that defects are captured (e.g., regression tests). Interesting, for traditional solution development environments, is the absence of a test management role which is normally expected to determine testing standards and policies, manage resources, provide oversight of the test process, coordinate and engage in stakeholder management and reporting. In some respects these tasks may be subsumed by other leadership roles within the team (e.g., Team Leader, Technical Co-ordinator, Solution Tester), but on the whole this situation reflects the lack of a separate test team function, relatively flat structures within the agile team and the broad skill sets of its members possess which enable a leaner approach to testing.

These remarks do not preclude the involvement of other roles in testing. In particular the Business Ambassador should encourage and support solution developers to integrate appropriate testing into their daily work.

7.6 Management Implications

A diversified approach to defining quality is recommended in order to take into account the differing perspectives of stakeholders. For example, whilst the solution team can identify with feature and performance quality attributes owing to their ease of automation, a business stakeholder may value serviceability or perceived user quality more. Accordingly, each quality dimension comes into focus at difference phases of solution development and although the DSDM approach to quality is broad, it should not be understood as a license to reduce quality to a single narrow definition. Equally important is the fact that quality operates at solution (i.e., quality control) and process (i.e., quality assurance) levels. Thus how quality is to be tackled should be evident in the Solution Architecture Definition and Development Approach Definition respectively supported by embedded testing practices (e.g., MoSCoW and risk based prioritisation), high visibility (e.g., test coverage and static code analysis displayed on communal screens) and reflective exercises (e.g., reviews and retrospectives). These are the indicators for an agile manager wishing to assess whether or not the team is truly "quality infected".

A review of the DSDM testing principles makes clear the implications for management of the importance of automated testing procedures [241]. In particular continuous integration environments embody the principles of fast failure and repeatable testing. Feedback loops (e.g., automated notifications to those culpable for failed tests) improve significantly the learning that can take place in this context. Together with prioritised and risk testing (e.g., the use of smoke testing practices to ensure that high priority tests are executed frequently relegating full testing to off-peak periods) this represents a potent combination for ensuring the early detection and rectification of defects.

It may also be worth considering investing in adequate project configuration management (see Chap. 9 for a discussion of this topic) in order to support development and to achieve deeper insights into quality and testing. Indeed, most IT solution development teams already employ version control systems for software assets and these often provide tools that help locate changes that are responsible for defects. A typical approach used by some distributed version control systems is the bisect algorithm which progressively subdivides a given range of change sets within the version control system applying a test case in order to determine the precise point in time when the test transitioned from success to failure. Furthermore, a mature integrated IT configuration management system wherein code assets are continually monitored (e.g., using static code analysis), test coverage is routinely recorded and operational incidents can be used to reliably link the solution development process to the quality of the deliverables it produces. Such a system ought to be capable of furnishing

answers to questions such as the effectiveness of test coverage or quality measures to reduce the number of operational issues, providing genuine process feedback far beyond what traditional performance metrics are capable of delivering. In spite of the success of agile testing strategies there remain challenges that need to be actively managed:

- Coping with testing in light of the scale of changes that a project may face can be a considerable challenge, especially when the need to rewrite existing tests is taken into account. Explicit prioritisation (e.g., MoSCoW or risk-based testing) is critical in this context as is an automated test environment that is capable of identifying where changes have led to new test failures.
- Calibration of testing and development activities refers to ensuring that estimation includes the total cost of delivery of a requirement. Often the immediate needs of solution development take centre stage during Timebox planning, overshadowing the necessary testing (e.g., unit, integration, system, regression). Inclusive and collaborative participation during planning, together with a stable team on which to build common understanding and experience, usually suffice to resolve this issue over the course of a few Timeboxes.
- Maintaining diversity of testing strategies helps improve the overall defect detection rate albeit at additional overall cost. An over reliance on specific forms of testing (e.g., unit testing) can seriously curtail overall test effectiveness. For example, studies have shown that unit tests achieve detection rates of approximately 25–30 % suggesting that they are only part of the overall solution [136]. Accordingly, testing within an agile team should retain elements of standard and exploratory practices, rather than becoming overly technocratic in nature.
- Dealing with regression (e.g., defects that arise from underlying changes) is a particular challenge in agile environments that embrace change and thrive on volatility. Automated regression testing (e.g., writing a test case tailored for a specific defect) will of course assist, though evidence points to the need to adopt a risk-based approach at the system level if regression is to be managed effectively [28, 137].
- The key challenge in integration testing is alignment with the overall flow of an agile project. This includes not only having components in a suitable state for integration (i.e., synchronization of current teams), but also accommodating long run periods for extensive and complex integration tests (e.g., tests running for several days or weeks). Mostly, this situation arises from the mistaken belief that everything must be completed first before integration can occur. In fact, this is precisely the problem that continuous integration originally set out to solve [177] wherein new components are continually added (e.g., perhaps using mock and stubs to "fake" incomplete implementations) to an overall system with varying degrees of testing (e.g., frequent high level prioritised tests and more comprehensive testing less performed less often) accompanying the integration.

Chapter 8
Risk Management

Abstract Though explicitly embedded into the DSDM Agile Project Framework, risk management in most agile methodologies remains a passive and implicit activity that can be misdirected and often misunderstood. It is telling that whilst most solution developers have little difficulty explaining which features they are working on (e.g., requirements) or to what level of quality they should be completed (e.g., acceptance criteria), few can comment on the capacity of their work to reduce (or exploit) project risk. These shortcomings can be addressed by applying agile risk management practices that embody those aspects of traditional project risk management which lend themselves to application in the spirit envisioned by the agile manifesto. This can include identification of key risk drivers that link back to wider programme risk management, acknowledgement of social and cultural influences on risk management along with risk tailoring of the underlying agile process. Thereafter operational risk management takes a moderately familiar form albeit with amendments that make it more conducive to agile environments.

8.1 Introduction

Risk management is concerned with the variability of outcomes in relation to project objectives the sources of which can often be traced back to one or more risk drivers that repeatedly appear in projects [71, 73, 184]. In spite of this, most agile methodologies frame risk in purely negative terms and limit its scope primarily to business and technical domains. Moreover, there is little guidance available with regard to the identification and treatment of risk and practically no recognition of the cultural attitudes towards uncertainty or the management of risk at the project or enterprise level. This suggests that those agile methodologies that lack a risk management framework may suffer from the following deficiencies:

- *Inability to make informed risk and reward decisions*. A central function of risk management is the recognition of threats and opportunities within a project and to balance the desire for reward against the risks incurred in its pursuit. Accordingly an understanding of the risk tolerance of a project (which is linked to the risk

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_8

appetite of the organisation) together with the nature of the risks encountered in a project is central to such decision-making.

- *Failure to identify appropriate risk response strategies based on risk exposure*. Risk exposure is a key determinant in the classification (and where appropriate prioritisation) of risks. The inability to recognise risk exposure may therefore impede the selection of an appropriate response. This point also encompasses the selection of agile techniques appropriate to the level of risk being managed.
- *Lack of oversight in risk monitoring*. Failure to engage in the monitoring of risk results in an inability to judge whether or not risk are being adequately managed. Team members ought to know how their activities are affecting project risk and how effectively and efficiently they are addressing that risk. This shortcoming extends to enterprise risk management through the failure to identify projects that have overstepped risk boundaries in relation to the overall level of risk that an enterprise is willing to accept or is capable of absorbing.
- *Poor understanding of when to engage in risk activities*. Lack of understanding or inconsistencies about the perception of risk means that the responses to risk events will vary amongst team members who fail to explicitly agree on appropriate controls and triggers. Elements such as risk compensation and other cultural influences also come into play at this juncture. Risk compensation refers to the shifting of risk elsewhere in light of risk treatment actions. For example the imposition of a speed limit on a stretch of road has been found to result in uptake of speeding just outside of the zone [3].

Understanding risk and how to manage it expands the awareness in an agile team already capable of articulating its commitment to working on business needs (e.g., Prioritised Requirements List), the relative importance of tasks (e.g., MoSCoW) and what constitutes done (e.g., acceptance criteria). Whilst many aspects of risk management may well be recognisable to those familiar with traditional risk management practices there are a number of features that distinguish the agile approach:

- *Cadence*. Whilst high-level risk analysis constitutes part of the project initiation (e.g., Feasibility Assessment and Foundations Summary), operational risk identification and analysis generally occurs at the start of each Timebox ensuring that risk management remains at the heart of planning and implementation throughout the project.
- *Collectivism*. Risk management is an activity that is engaged in regularly by all team members and it is this diversity of involvement that strengthens the overall process. In contrast to traditional approaches, there is no risk manager to whom risks must be escalated and who must coordinate an appropriate response. Instead, whilst complex projects may well engage the services of risk specialists or managers this is not a license to abdicate responsibility for risk awareness. Rather there is always a collective sense of self-organisation and responsibility in the identification, analysis, treatment and monitoring of risk.
- *Transparency*. Risk artefacts are made purposefully visible so that all participants understand the systemic level of risk and the distribution of risk and reward. In

addition, progress towards tackling risks is displayed using Burndowns to provide continual feedback to the team.
- *Agility*. Whilst the classical response strategies of traditional management are retained, Agile offers a more expansive set of practices and techniques that may be brought to bear on risk.

Throughout this chapter risk management will be discussed in high-level terms often with reference to DSDM though the views of the wider agile community are also incorporated. Operational matters, whilst alluded to, will not be delved into in great detail as these are covered adequately elsewhere in the literature [48, 71, 184, 185].

8.2 Definition of Risk

Risk is understood in the context of projects and defined as uncertainty that has an impact on project objectives [109] and is consistent with the definition found in Enterprise Risk Management [14, 132, 181] and related practices [5, 110, 213] including its application to IT [129, 130]. This is important since project risk should not adopt an "island mentality" and ought be managed in the context of wider enterprise goals, consistent with their view on risk attitude, appetite and tolerances. Risk is commonly described in terms of its components of likelihood and impact which are referred to collectively as risk exposure. Note that in the literature it is not uncommon to use the terms risk and risk exposure interchangeably. For example a statement such as "the risk of at least one financial loss of more than one million dollars within the next six months" incorporates both impact and frequency, a common proxy for likelihood, as well as proximity. Risk exposure is sometimes formally expressed as a co-ordinate pair, or the individual components may be converted into numbers and the exposure computed as their product. Particular care should be taken, however, when calculating exposure in this fashion, as converted numerical values may in fact be nothing more than ordinals for which a mathematical product carries no semantic meaning whatsoever, though its magnitude may nonetheless have some indicative character about it. This is commonly the case when using T-shirt sizing to rate risks with a scoring to convert them into values (e.g., "S", "M" and "L" may be assigned a values of 1, 2 and 4 respectively).

Risk is often equated with statistical uncertainty. Uncertainty may or may not be something that can be accurately measured and thus it may help not to make the differentiation between "risk" (knowable in probabilistic terms) and "uncertainty" (unknowable randomness) as described by [145] who took a mathematical standpoint on the matter. Later when addressing risk assessment the argument will be made that estimation of uncertainty is inherently a subjective matter and since there is rarely recourse to strictly probabilistic means of judging risk, this distinction is moot for all practical purposes. Inherent risk is related closely to two other forms of risk. The first is residual risk which refers to that risk which remains leftover once an

activity has been undertaken to treat the inherent risk and the second is secondary risk that arises on account of the treatment of the inherent risk.[1] Thus what emerges is a picture of risk as a web of causality wherein identification and treatment can be dynamic and complex. It is important to understand though that not all forms of uncertainty are relevant and that risk in this context is understood to be related only to uncertainty that has a bearing on objectives. Risk can have upside (positive) as well as downside (negative) consequences. It is commonly the case that negative risk is implicitly implied and, as indicated earlier, this appears to be the prevailing view in the agile literature. However, opportunity (positive risk) can manifest itself in many forms ranging from operational issues (e.g., the need to cater for higher than anticipated demand for a service) to strategic matters (e.g., extensibility of product into new market segments).

Consider the opportunity in a software product where its use may exceed expectations by tenfold. Since this is a desirable outcome with considerable financial upside, one or more measures might be considered to try to promote this outcome. These might range from technical measures (e.g., improving the algorithms to ensure higher scalability), to infrastructural amendments (e.g., switching to a load balancing topology) and deployment related actions such as increasing awareness of the product (e.g., marketing). Were all enacted, it would be likely that the risk would be realised and the lost opportunity of still further use of the product would be reduced as the product begins to saturate the market.

One comment on risk management terminology in respect of positive risk that is worth making at this point is that the exploitation of positive risk results in a reduction of uncertainty in relation to the opportunity that would otherwise have been lost (i.e., risk exploitation reduces residual risk). In summary, therefore risk is uncertainty that is of relevance and which can result in upside or downside consequences. To employ an metaphor, the outcome of a horse race may well be uncertain but is irrelevant until a bet has been placed at which point the uncertainty directly affects the making (positive) or losing (negative) of money thus constituting a risk.

8.3 Agile and Risk

Although frequently mentioned, risk tends to be treated in a rather narrow and implicit manner in the agile community. Later in this chapter the primary sources of IT project risk are identified as project (approach), schedule, supplier, people, requirements and technical, yet consistently agile methodologies appear to focus almost exclusively

[1] In both cases anticipated activity made be planned and the assessment of residual or secondary risk may reflect the outcome of a statistical model rather than actual undertakings.

on requirements and technical risk with few (e.g., DSDM) appearing to be aware of other sources of risk. Equally there is a prevailing perception that risk must inevitably be understood in terms of project threats (negative risk) and thus the opportunities (positive risk) presented in agile projects are inadequately managed. Yet nowhere more so than in IT do opportunities present themselves through the value enabling nature of projects, the manner in which they are delivered and the operations and service delivery aspects that provide ample scope for exploiting synergies, stream lining processes and sustaining value creation [129]. Given the mantra of "embrace change" that pervades the agile community and in light of the enabling nature and value creation potential of projects, it seems ironic that such (positive) risk evaluation does not feature more strongly in agile projects, i.e., "embrace risk". Fortunately, wider appreciation of risk is not entirely forgotten and is sometimes used as one of several factors used in the prioritisation of task [52]. This notwithstanding, the agile community on the whole does appear to be lagging behind the risk management community, which for some time has understood its function as informing decision-making in relation to the balance between risk and reward.

Perhaps the most commonly cited deficiencies of risk management in Agile include the lack of an explicit definition of risk, the focus on development aspects of risk (i.e., requirements gathering and technical implementation) without consideration of risks elsewhere in the solution development process, the nature of responsibility for risk, the recording and monitoring of risk, acknowledgement of the environment in which the project takes place together with organisational attitudes to risk [1, 194]. Indeed many of the notions that ordinarily arise in the context of project risk management such as those described in the PMI Body of Knowledge and its more risk focused publications, the Management of Risk framework which relates more to PRINCE2® and other sources, are simply nowhere to be found in agile project management [109, 195, 210, 211]. Given the growth in recent years of enterprise risk management together with their application to IT this attitude to risk management seems parochial [14, 129, 132, 181].

One of the most influential early references to the embedding of risk management in iterative software development lifecycle was the *Spiral Model* [29, 30], wherein a process model was described that actively seeks to identify and resolve risks and use these to influence the evolution of a solution from requirements through to operations. Though inherently iterative in nature, it was argued that this was not limited to a specific type of solution development process. Crucially the point was made that risk determines both the level of effort (e.g., perform product testing only to the extent that it reduces risk to an acceptable level) and the degree of detail (e.g., apply more design effort to innovative elements of a project) that risk related practices employ. These ideas were developed further into the notion of risk-driven architecture which advocates that the selection and application of techniques be motivated by risk and be commensurate with the risk of failure or success [87].

One aspect of risk that does receive some attention in agility is the balance of risk and delivery of value to customers when prioritising tasks especially when doing so also takes into account other factors (e.g., cost). For example the strategy of addressing high value and high risk, high value and low risk and then low value

and low risk tasks in that order, whilst entirely avoiding low value and high risk tasks is commonly advocated [52]. The argument is that working on high value and high risk tasks first eliminates significant risk early on. This approach treats risk as a facet of a task which might, however, prove to be too limiting. For example some of the risks which a project must contend with are not inherent in the execution of specific tasks, but rather in the circumstances surrounding that execution and might otherwise be considered part of a project governance profile (i.e., the effective and efficient deployment of resources towards the achievement of the goals of the enterprise).

There are strong parallels between risk management and group learning which tackle the central issue that all too often deviations from what ought to have been done are noticed too late in a project (e.g., at point of integration or deployment) as described in [48]. In particular when attempting to establish whether or not the correct solution with appropriate technical design is being delivered at a reasonable cost by the right people, it is necessary to employ a variety of strategies revolving around business involvement, prototyping, creation of walking skeletons and architectural spikes, basing estimation of experiential sampling, ensuring that the team performance is boosted by early victories and establishing a safe environment with which to become acclimatized to the project through to structuring iterative development into learning, refinement and finishing phases. There is a resonance in such thinking with the DSDM approach (e.g., role of the Business Ambassador, practices such as Prototyping and the Timebox structure as terms of Investigation, Refinement and Consolidation) though other elements are clearly at odds (e.g., the targeting of simple, rather than high value, items in order to enable the team first to overcome social risks conflicts with MoSCoW prioritisation though arguably this should be described as a tactic rather than a strategy).

8.4 Cultural Attitudes to Risk

Risk assessment is not an entirely rational probabilistic activity, wherein data is precise and models accurately reflect the realities of the risk scenario. Indeed, experience suggests that this is seldom, if ever, the case but rather that assessments are largely a subjective and even visceral affair. One approach is to attempt to understand risk attitudes (rather than employ cultural typologies that are derived from underlying belief systems [120, 258] though such theories been critiqued on the grounds that "empirical evidence for this theory is … sparse" [3, p. 38]) for which the following classification may prove to be of interest [109]:

- *Risk-averse*: Preference of secure payoffs, common sense and facts over theories. Propensity to over-react to threats and under-react to opportunities.
- *Risk-seeking*: Preference towards speculation and unafraid to take action. Propensity to underestimate threats and overestimate opportunities.

- *Risk-tolerance*: Indifference towards uncertainty that lends itself to reactive rather than proactive measures. Propensity to fail to appreciate importance of threats and opportunities alike.
- *Risk-neutral*: Impartial attitude towards risk and act in the interests of significant benefits. Propensity to focus on the longer term.

These observations [190] are linked to the notion that risk attitude is influenced by the "triple strand" of conscious factors (e.g., visible and measurable characteristics), subconscious factors (e.g., heuristics) and affective factors (e.g., visceral feelings). The relevance of these studies for agile risk management is that they appear to indicate that it is at the point of choosing a risk response that risk attitude plays a crucial role (i.e., tendency to engage in a risk or withdraw from it). Thus there must be an appreciation within a team that different members may hold fundamentally differing views towards risk and that the conflict that arises when assessing risk ought not be judged entirely in terms of the rationality of opposing arguments. The emotionally intelligent agile project manager ought, therefore, to invest some time understanding the risk disposition of the members of their team.

When considering organisational and national culture, the "Uncertainty Avoidance" Index (UAI) as detailed along with other dimensions in [117] may be of interest. Uncertainty avoidance is defined as "extent to which the members of a culture feel threatened by ambiguous or unknown situations and have created beliefs and institutions that try to avoid these" [118] and is described in more detail in the following terms:

> The uncertainty avoidance dimension expresses the degree to which the members of a society feel uncomfortable with uncertainty and ambiguity. The fundamental issue here is how a society deals with the fact that the future can never be known: should we try to control the future or just let it happen? Countries exhibiting strong UAI maintain rigid codes of belief and behaviour and are intolerant of unorthodox behaviour and ideas. Weak UAI societies maintain a more relaxed attitude in which practice counts more than principles. [118]

Just how widely UAI varies is illustrated in Fig. 8.1 [118], which was based on data collected between 1967 and 1973 and periodically updated since then, to depict the attitudes towards uncertainty amongst the G-20 of major economies.[2]

Since uncertainty avoidance is related to risk attitude it has a bearing on how individuals may perceive and attempt to control risk within projects and organisations. In particular in the light of risk classification and its influence on choice of risk responses it could be argued that risk propensity is the dominant characteristic in risk behaviour [244]. These considerations become particularly acute when dealing with teams distributed across several cultures which is likely to be the case in geodispersed agile teams that are commonly featured in enterprise forms of agility.

[2] G-20 comprises of nineteen member states plus the European Union which is excluded from Fig. 8.1 owing to lack of aggregate data.

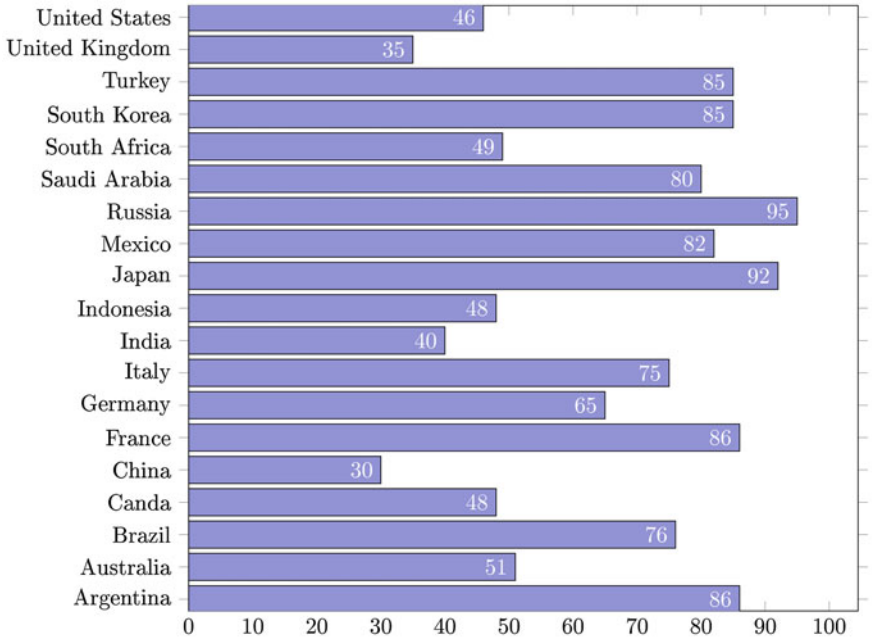

Fig. 8.1 Hofstede Uncertainty Avoidance Index for the G-20 Member States (excl. European Union). Published with kind permission of © Alan Moran 2013

8.5 Sources of Risk

It may be appropriate to identify the key risk drivers for the sector in which an organisation and its projects operate. This may fall under the remit of enterprise risk management and constitute a project management office function in relation to individual projects. For example, numerous research studies have identified various generic categories of IT project risks [12, 31]. Table 8.1 taken from [184] describes a list of generic IT risk drivers that may help as a starting point. Note that not all risk of relevance to a project needs to be tackled at that level but can be delegated up to the programme level or to a specialist function. For example, currency risk for projects that invest heavily in off-shore sourcing could call upon a treasury function to assist in mitigating risks (e.g., purchase of currency options).

Table 8.1 Common Risk Drivers for IT Projects

Risk Driver	Description
Business Risk	All risks relating to functional requirements and the changing nature of requirements in general. Issues of user acceptance might also be relevant to this category depending on the nature of the project. Remaining non-functional requirements not already accounted for under technical risk may be included here (e.g., usability, security)
Technical Risk	All risks relating to architecture, design and infrastructure of the proposed solution. This risk is not limited to the developed solution but also encompasses dependencies (e.g., shared libraries) together with the estimation of hardware and software dimensioning and capabilities. Typically the majority of non-functional requirements are included here (e.g., maintainability, scalability, performance)
Schedule Risk	All risks arising from the scheduling and timing of activities (including the release planning of increments) and the financial cost consequences thereof (e.g., net cash flows, investment decision-making)
Project Approach Risk	All risks relating to the effectiveness of the project management methodology, levelling of resources and the management of project complexity. Depending on circumstances it may also be appropriate to include risks relating to management support for the project
Supplier Risk	All risks relating to external sourcing including consulting and delivery of components (incl. timeliness, conformity, quality). It is not uncommon to subsume risks relating to the stability, continuity and capability of suppliers into this category
People Risk	All risks related to the level of skills in the team and expectations of abilities. These risks are often impacted adversely by other risks as impending project deadlines place greater stresses on staff

8.6 Enterprise Risk Management

Enterprise Risk Management (ERM) extends the traditional risk management focus which is chiefly concerned with specific domains (e.g., projects, IT, security, health &safety) to a more holistic view encompassing the entire organisation [14, 55, 129, 130, 132]. The scope of ERM supports alignment of risk management practices when tackling complex objectives that are not under the remit of a single project team or department (e.g., mergers and acquisitions, large scale data migrations, IT security and compliance). It can also support the use of a common risk language throughout the organisation (e.g., common scales to assess risk impact) thereby making localized risks more comparable and enabling the aggregation of risks to better understand risk distribution [130]. From an agile perspective the most important point to make is that whatever occurs at the project level ought to have some bearing and linkage to wider enterprise risk management concerns (e.g., terms of reference of project risk expressed as key risk drivers, aggregation of risk to the programme level).

That ERM seeks to improve the "the management of increasing risk mitigation costs and the success rate of achieving business objectives" [213, p. 3] by helping reduce unwanted performance variability reflects the widely held perception that good governance leads to better performance and that responsibility for risk lies with the board of directors [53]. Indeed the board has a duty to achieve an understanding of risk and communicate its significance throughout the organisation. Risk management is repeatedly cited as a central component of governance that features in a variety of IT frameworks. As a result well risk managed organisations better understand the balance on the one hand between risk and reward, and on the other the capabilities of the organisation and its ability to absorb losses or fail to capitalise on opportunities. This in turn builds investor and stakeholder trust who retain confidence in the organisation to weather bad times and cope with changes in the future.

To better understand the nature of ERM it helps to take a look at the financial statements of major listed companies wherein a description of the potential impact of risks on strategic objectives can be found. For example, IBM cites the need to be able to "better monitor, predict and manage risk to build trust and value amidst uncertainty, by having confidence in their data, risk exposures and ability to make risk-aware actionable decisions" whilst mostly focusing on managing financial (e.g., financing, credit on receivables and currency fluctuations), supply chain and security risks [125]. Such openness demonstrates an awareness of the market and a willingness to adapt. Far from unsettling investors such admissions make clear that these matters have the full attention of senior management and that measures are being considered to address them.

In light of financial scandals and corruption in the US dating back to the mid-1970s the Committee of Sponsoring organisations of the Treadway Commission (COSO), a joint venture comprising of the Institute of Management Accountants (IMA), the American Accounting Association (AAA), the American Institute of Certified Public Accountants (AICPA), the Institute of Internal Auditors (IIA) and the Financial Executives International (FEI), set about defining standards covering governance, ethics, risk management and financial reporting. Further scandals involving Enron and Worldcom resulted in the passing of the Sarbanes-Oxley (SOX) Act in 2002 which was followed in 2004 by the COSO ERM Integrated Framework standard compliance which is generally accepted to be in accordance with the SOX legislation. The following broad and comprehensive definition of Enterprise Risk Management (ERM) describes it as:

> a process, effected by an entity's board of directors, management and other personnel, applied in strategy-setting and across the enterprise, designed to identify potential events that may affect the entity, and manage risk to be within its risk appetite, to provide reasonable assurance regarding the achievement of entity objectives. [55]

Table 8.2 Comparison of Enterprise and Project Risk Management

Enterprise Risk Management	Project Risk Management
Wide scope that requires the commitment of senior management and the engagement of the entire organisation	Narrow scope that chiefly concerns the project team and its stakeholders through direction by senior management may occur
Strategic focus on the objectives of the organisation	Tactical focus on project objectives
Concern related to the tangible and intangible assets of an enterprise that underpin its business model	Concern is limited to the project as a change vehicle
Governance function with responsibility for oversight of the management of risk that faces an organisation	Governance restricted to the level of risk management process (e.g., effectiveness, compliance) in relation to project objectives

Without dwelling further on details it suffices to say that there exists at the corporate level a mature understanding of risk in the context of which project risk management must find itself. This parallels the relationship between agility and strategic management alluded to elsewhere and suggests that concepts such as agility, risk and strategy ought not to be understood as incompatible terms but rather find expression at many different levels in the organisation. Table 8.2 provides a high-level comparison of enterprise and project risk management.

8.7 Agile Risk Management

The foundation of agile risk management (see Fig. 8.2) begins with the definition of project objectives and context and maps these in terms of organisational risk management. This enables the project as a whole to be risk assessed alongside other similar projects in order to determine if its proposed benefits warrant the level of risk seen from the programme perspective. Next, identification of risk drivers and determination of appetite is addressed in the pre-project phase though it may be returned to later if project circumstances alter significantly. Thereafter it is important to consider how the agile process being employed need to be tailored in light of the risk environment of the project. This is a project (rather than methodology) specific undertaking since tailoring is an activity that is highly dependent on project parameters and circumstances. The operational aspects of risk management involve amendments to traditional risk management practices (i.e., identification and analysis of risk, determination of response strategies concluding with treatment and monitoring of risk) to make them more conducive to agile environments.

Agile risk management is guided by three principles which work in concert with values, such as openness, respect and courage, inherent in most agile methodologies

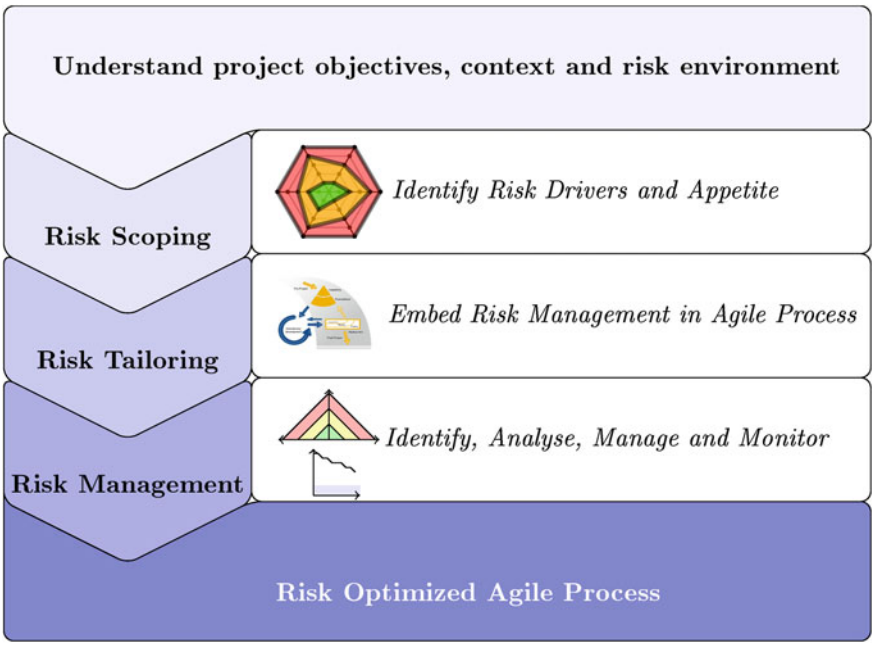

Fig. 8.2 Agile Risk Management Process Overview. Published with kind permission of © Alan Moran 2013

and reflect features like communication and collaboration found in the principles of the agile manifesto:

1. *Transparency*. All risk related activities and artefacts should be visible to everyone in the team at all times. It is proposed that these be placed next to other agile tracking and reporting tools (e.g., Burndowns, Team Board) or in a central project area (e.g., information radiator) and that team members be permitted to add to or annotate them. This practice is referred to as risk walling and it means that someone outside of the project could walk into the project area and immediately assess the risk situation without having to ask or interrupt team members.
2. *Balance*. Risk management is all about balancing risk and reward and finding ways of generating the same level of value with a lower level of risk. It should therefore be obvious which requirements bear the most risk and how the work of individual team members contributes to risk mitigation by either reducing threats or exploiting opportunities.
3. *Flow*. Risks are unavoidable in projects but understanding them and knowing how to deal with them enables the project to continue without serious disruption. For example, contingency plans agreed in advance make sure that should accepted risks materialize, the team knows what to do, how to prioritise their work and is not interrupted with replanning or crisis activities.

There is a natural mapping between these principles and those of DSDM. This includes the need to ensure a common understanding of risk and reward, linkage of risks with business objectives, appreciation of the influence of the distribution of risk on deliverables, ensuring diversity of input into risk identification exercises, embedding risk consideration into MoSCoW prioritisation and demonstrating clear oversight of risk and its management at the Timebox level [185].

The suggestion that risk is a responsibility carried by all project members endears itself to the communal spirit of co-ownership and good citizenship that is expected of agile teams. This notwithstanding, the role of risk manager is sufficiently important for it to be assigned to an individual in the interests of ensuring that there is process compliance and that there is accountability at the process level for the effectiveness and efficiency of risk management activities. In DSDM it is expected that the Project Manager assumes this responsibility but that this individual delegates identification, analysis, treatment and monitoring to the team relegating their own role to a coaching and supportive function. Whilst the Project Manager may well maintain risk related artefacts (e.g., risk log, risk burndown), ownership remains with the team and full visibility must be maintained throughout. Needless to say, other risk specialists (e.g., quantitative or domain risk experts) may also be involved in an advisory capacity depending on the circumstances or complexity of the project.

The clarification of project objectives, context and environment alluded to earlier (see top of Fig. 8.2) takes place in the enterprise risk management context of the organisation and involves establishing an understanding of what the project is endeavouring to achieve and how much risk the enterprise is willing to tolerate in pursuit of these goals. This occurs during project initiation (e.g., somewhere between

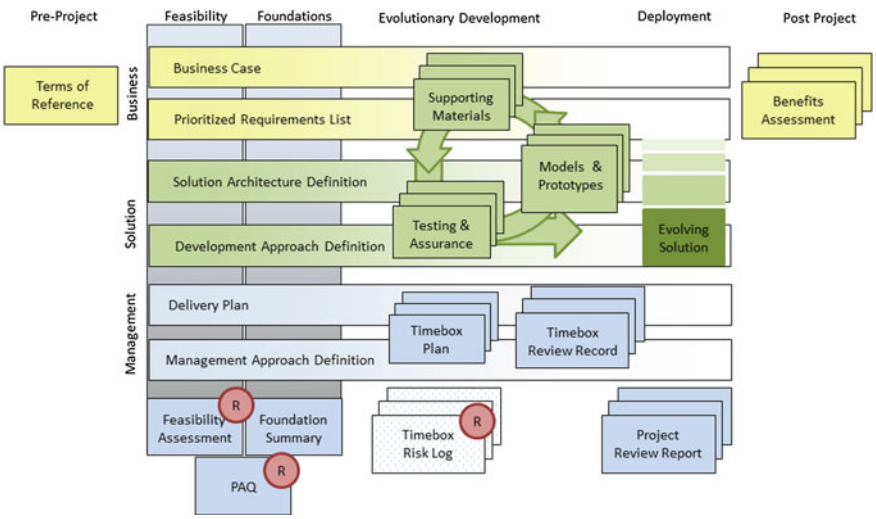

Fig. 8.3 Role of Risk Management in DSDM Products. Published with kind permission of © Alan Moran 2015

the Pre-Project and Foundations phases) and requires that the project objectives be known which in turn provide the basis of what constitutes risk (Fig. 8.3). Next the context of the project must be established since it is this that might exempt the project from ordinary controls (e.g., allowing more risk to highly innovative product launches) or otherwise constrain it (e.g., redefinition of project scope if deemed to be too risky). At this point in the process must merely be noted and any decisions relating to the risk nature of the project be consciously taken and approval, where appropriate, be sought. The point here is not to invent bureaucracy but rather to be aware of projects that are being proposed which might be at odds with enterprise risk tolerance (e.g., it is not appropriate in most industries to permit projects to engage in highly risky undertakings that could damage the reputation or financial standing of the organisation). Thus the risk environment which also encompasses regulatory aspects and the capacity and willingness of the organisation to engage in enterprise risk management practices may also impact at the programme or project level.

An understanding of risk drivers helps not only clarify high-level attitudes towards and tolerance of risk but also assists risk identification by raising awareness of common sources of risk and ideally this should occur at the programme level. One approach is to grade each risk driver in order to reach a consensus concerning the acceptable upper and lower risk thresholds. The upper risk threshold is that level beyond which urgent and immediate risk mitigation activity would need to be initiated. This embodies an uncomfortable level of uncertainty with which the organisation would have difficulty coping with. The lower risk threshold is that point below which the risk is considered negligible and scarcely needs further monitoring. This is the comfort zone and the realm of daily business. The region between the two thresholds represents risk that ought to be monitored and acted upon at an appropriate time if deemed necessary. When grading risk drivers, a simple scale comprising of five to seven points is often adequate. The grades should be expressible in business terms and be understood by all participants. Table 8.3 is a simple five point scale that describes the range of risks that might be encountered in the technical risk category. When determining thresholds, always bear in mind that risk can have upside as well as downside. Thus technological uncertainty could be considered as much an opportunity as a threat and this may be reflected in the willingness to embrace risks if there is a perception that this may bring advantages. Therefore care should be taken not to frame the grading system in unduly negative terms. Insofar as this is possible a standardised set of drivers and scales ought to be used at the programme level and embedded in this should be the consciousness of enterprise risk management environment. Isolated individual projects, however, need not be subject to this constraint since standardisation brings no benefit.

Once a graded list of risk drivers is available it is time to reach a consensus concerning the upper and lower risk thresholds for each of the risk drivers within a specific project context. For example in Table 8.3 an enterprise might feel that generally speaking the levels “Terra Firma” and “Cautious Explorers” hardly constitute concern though might be conservatively minded enough to consider the risk surrounding “Early Adopters” to be a step too far unless the project is of such special significance that this level of risk is warranted.

Table 8.3 Sample Five Point (Descending) Scaling for Technical Risk

Scale	Description
Market Makers	Highly innovative and ground breaking technology that requires new and perhaps unproven technologies or application of existing technologies in unanticipated fields
Early Adopters	Inroads into technologies that have seen some industry use though a significant effort must be expended to become capable and effective in their use (e.g., tool support might be immature)
Forward Movers	Significant technical innovations on several fronts where industry best practices and reference implementations already Exist. Technical practices may require interpretation and adaptation to the organisation
Cautious Explorers	Mainstream technical implementations that require some evaluation of new technologies integrated into existing frameworks and platforms. Infrastructure is established but training in its use might be required
Terra Firma	Well tried and trusted technologies in which the organisation is highly invested. Tools and processes are well known and adhered to

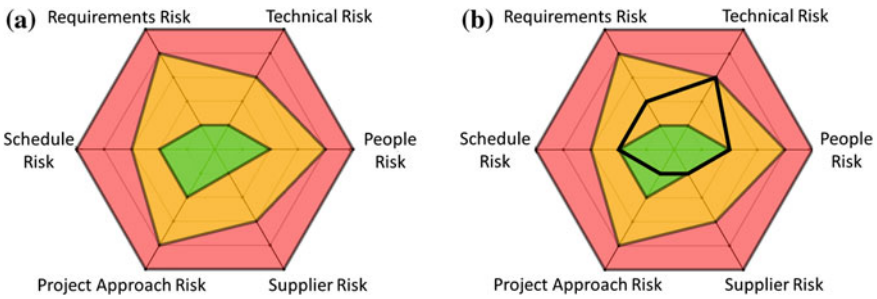

Fig. 8.4 Risk Driver Maps. Published with kind permission of © Alan Moran 2013. **a** Enterprise risk driver map. **b** Project risk driver map

An enterprise risk driver map is created by connecting together all the upper respectively lower risk threshold points of each key risk driver and marking the region above the upper limits red, the region below the lower limits green and the remaining region yellow (see Fig. 8.4a). Thus a risk driver map is a visual indicator of the attitude towards risk at the enterprise level. The value of a risk driver map becomes clearer when the personal attitudes towards risk of the project team members are taken into account. Willingness to take more risks or the adoption of a risk averse stance may prove to be at odds with the project risk appetite and thus it becomes necessary to communicate what constitutes acceptable and unacceptable risk. It is at this stage in the process that the project manager should take the time to understand the personal risk attitudes of team members.[3]

[3] For example, DSDM promotes the use of a Project Approach Questionnaire that could be adapted for this purpose.

Plotting of individual projects on the risk map (see Fig. 8.4b), can be used as the basis for assessing whether or not to proceed with individual projects or to decide if exceptions need to be considered. This may also stimulate the application of tools consistent with risk-driven approaches that advocate the selection and application of techniques motivated by risk and commensurate with the risk of success or failure. The value of the consistency of an enterprise risk driver map is that it enables knowledge gained in assessing risk appetite from completed projects to be transferred over to new ones. Indeed if an enterprise risk management framework (e.g., ISO 31000, COSO or Risk IT [129, 132, 181]) is already in place, then there should be some form of linkage between project risk appetite and the wider corporate attitude towards risk.

The next stage of the agile risk management process (see risk tailoring in Fig. 8.2) concerns project specific amendments to the agile methodology which can only be done in light of an understanding of the risks facing the project already identified in the risk scoping stage. Up to this point in the process the agile methodology in use has not been considered, however, it is now necessary to begin looking at the specifics of the chosen methodology. For example, a recommended tailoring of DSDM is illustrated in Fig. 8.5 which contrasts the standard process based on a structured Timebox (see Fig. 8.5a) with a risk tailored variant (see Fig. 8.5b).

This helps clarify when and how often specific activities are expected to take place. Indeed this has an impact on the manner in which risk management is performed (e.g., does the majority of planning occur at the incremental or iterative level? when should risk assessment activities take place?). When considering where to place risk management activities it is important to consider the state of available information. For example, it may be tempting to conduct a risk analysis at the increment level but if there is insufficient information to evaluate risks properly then it is more appropriate to perform this analysis at the iteration level. Agile techniques themselves (e.g., prototyping, continuous integration, refactoring) should also be considered risk management tools though how they are deployed depends on the necessary level of

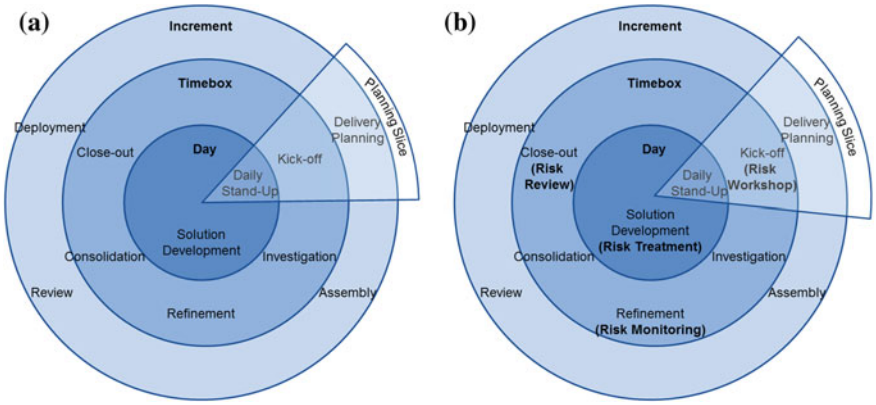

Fig. 8.5 Risk Tailoring of the DSDM Process. Published with kind permission of © Alan Moran 2015. **a** DSDM Agile Chart (Structured) **b** DSDM Agile Chart (Risk Tailored)

information required together with the desired frequency and intensity of the activity. In particular the decision to undertake such activities is often task related and therefore it would not be appropriate to mandate their use during methodological risk tailoring.

The operational details of agile risk management build upon traditional project risk management practices albeit with a number of minor changes that promote collectivism, agility and transparency [184, 185]. Thus, for example, the planning during the Kick-off subphase of a Timebox may be augmented with a risk session wherein risks are identified, analysed and courses of treatment determined that feedback into the tasks which must be undertaken. Risk identification is the activity of determining the main uncertainties that could have a plausible and material effect (positive or negative) on the project objectives. However, confusion often arises when attempting to distinguish between risks and non-risks, in particular causes and effects and what is so pernicious about such misunderstandings is that it often directs risk management activities towards actions that fail to address the underlying risks. A simple technique for identifying risks that is amenable to agile environments (illustrated in the left pane of Fig. 8.6) is to ask *what* may occur (i.e., effect) and then follow-up with questions concerning *why* this might be so (i.e., risks).

> To understand the confusion between risks and effects, consider the migration of a website from one server to another. Often its non-availability is cited as a risk, however, this is the *effect* of an unsuccessful migration and is not the origin of uncertainty. Closer inspection may reveal why the website was unavailable after the migration (e.g., uncertainty in DNS configuration or whether or not the two hosts were in fact configured the same). These are the risks that require treatment (e.g., learning how to configure DNS or checking that both servers are set up in the same way).

Thereafter risks can be recorded in a risk log for assessment in terms of impact and likelihood (collectively known as exposure) as depicted in the lower middle pane of Fig. 8.6. The important point to bear in mind here is that the purposes of analysis is on the one hand to prioritise risks and on the other to be able to use exposure to determine a risk response strategy (see Table 8.4 for details) based on Fig. 8.7 which has been adapted from enterprise risk management for project context purposes. Where there is no consensus on a specific risk assessment, then a range estimate should be employed (as indicated by the bar in Fig. 8.7). These points are central since it is precisely at this stage in the agile risk management process that the social and cultural influences are at their greatest (i.e., individuals tend to instinctively make an assessment and rationalize it later). Accordingly discussions of risk exposure are permitted to be translated into ones concerning actionable behaviour (e.g., should a risk be mitigated or avoided). Thus risk assessment is not the time to indulge in stochastic modelling that creates a false sense of accuracy but rather to engage in action oriented dialogue that ranks risk activities in relation to the business value of the items to which they are associated thereby establishing a bond between risk and reward.

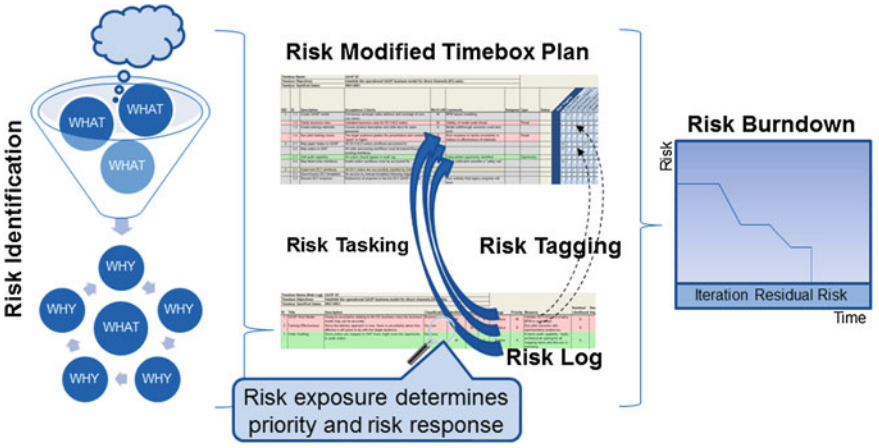

Fig. 8.6 Operational Risk Management. Published with kind permission of © Alan Moran 2013

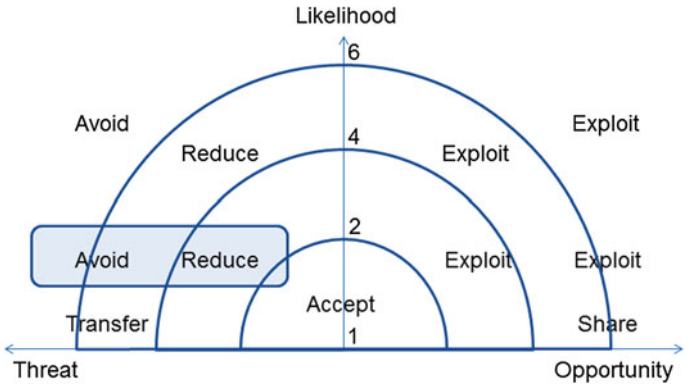

Fig. 8.7 Mapping of Risk Exposure to Risk Response Strategies. Published with kind permission of © Alan Moran 2015

Based on the chosen risk response strategy there are a number of options available that include removal of an activity that gave risk to the risk from the Timebox plan, adding tasks to the plan to mitigate, exploit, transfer or share risks or accepting that a risk may occur and determining a contingency plan for tackling it should it materialize. However, there is another option uniquely available to agile risk management that involves the association of agile techniques with tasks (or classes thereof) specifically for the purpose of risk treatment, a practice referred to as risk tagging [184, 185]. Thus, for example, all tasks related to the development of a new transaction processing algorithm may be subject to pair programming and intensive test-driven development techniques and accordingly such tasks may bear a tag or other form of identifier on the Timebox plan in order to remind the team of that decision. The translation of risk log items into actionable behaviours is depicted in the upper middle pane of Fig. 8.6.

Table 8.4 Classical Risk Response Strategies

Risk Strategy	Description
Accept	Appropriate for low risk exposure. No specific action will be undertaken to mitigate or manage the risk. Instead a contingency plan will be developed to tackle the eventuality that the risk may be realised
Reduce	Appropriate for medium negative risk. Actions will be undertaken to reduce either the impact or the likelihood of occurrence of the event
Exploit	Appropriate for medium to high positive risk. Measures will be undertaken to increase either the frequency or the impact of the event
Share	Appropriate for low frequency high impact positive risk where the efforts to manage the risk alone may not be warranted owing to the relative unlikelihood of it occurring. Measures will be undertaken (with others) to increase either the frequency or the impact of the event
Transfer	Appropriate for low frequency high impact negative risk where efforts to contain it may be beyond the capabilities of the team. Measures include the transfer of the risk to a partner capable and willing to treat it (e.g., out-sourcing to a specialist)
Avoid	Appropriate for high negative risk. The activities that give rise to the risk will not be undertaken

Finally the monitoring of risk involves assigning numerical scores (often loosely based on risk exposure as suggested by the values assigned to the bands of Fig. 8.7) to individual risks which is maintained as a cumulative sum over all risks. Risk is reduced by one or more of the following conditions (the effect of which is to burndown the risk by the difference of the related inherent and residual risk):

- Completion of risk related tasks. These recognise efforts taken to address the underlying risk (e.g., exploit or reduce).
- Decisions taken to affect a course of action in relation to a risk. These typically involve the reduction of inherent risk to residual risk in acknowledgement of a proposed measure (e.g., transfer or share).
- Decisions taken to perform tasks in a specific manner. This acknowledges the contribution of agile technique to the reduction of risk.
- Cessation of activity that gives rise to a risk. These concede that the risk invoked by the activity is infeasible or uneconomical to bear or share (e.g., avoid).
- Expiry of a risk. Recognition of activity that having been undertaken causes the source of the risk to dissipate.

Burndown charts, such as Fig. 8.8, are a familiar sight in the agile landscape that are used for a variety of purposes (e.g., tracking remaining effort). There are, however, a couple of points to bear in mind concerning risk burndowns. The first is that whilst a monotonically decreasing curve between risk assessment workshops is to be expected, the reality is that risk treatments often trigger secondary risks that may cause the curve to increase rather than decrease. Secondly, there exists an inherent systemic risk (e.g., comprising of cumulative residual risk, risks related to accept

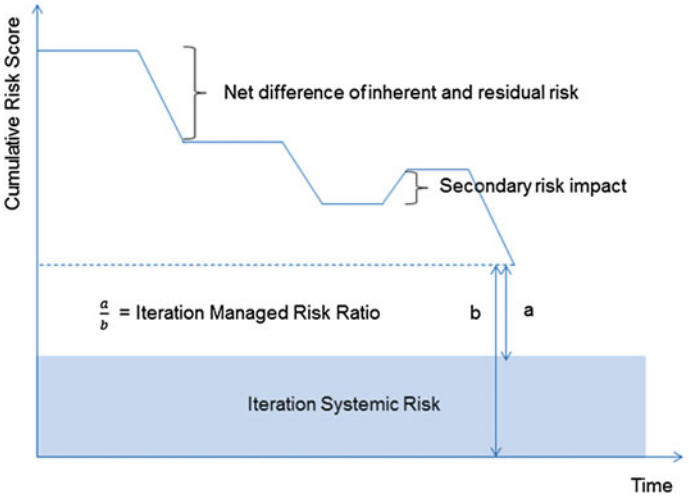

Fig. 8.8 Illustrative Timebox Risk Burndown Chart. Published with kind permission of © Alan Moran 2015

strategies) that cannot be entirely eliminated and must be acknowledged. Typically the iteration residual risk consists of:

- Risks associated with accept strategies where it has been decided to rely on a contingency plan in the event that the risk occurs.
- Residual risk of tasks where completion of a task does not entirely eliminate the risk (e.g., scalability measures to cater for unanticipated demand up a predefined level beyond which the risk of additional demand constitutes residual risk).
- Risk over which there is an element of loss of control arising from the action undertaken that persists beyond completion of the task (e.g., transfer or share strategies where a liability is retained giving rise to residual schedule risk).

Iteration residual risk is therefore a reminder to all project members that risk cannot be entirely eliminated and is a visual depiction of the extent to which unmanaged risk resides in the project. High iteration residual risk is a matter of concern and should be examined in the context of the project and overall enterprise risk appetite.

As a final note, the visualization of risk not only increases awareness within the team but also enables those less familiar with the technical details to frame questions appropriately and engage in meaningful discussion regarding the management of risk. Consider, for example, Fig. 8.9, which depicts a requirements map in which the top row are requirements and subsequent rows are their breakdown within respective columns in terms of individual tasks. Marked as dashed and thatched boxes are those tasks that relate to the treatment of negative and positive risk respectively. Finally several tasks are tagged with symbols (e.g., diamond, star and cross) indicating the application of a specific agile technique (e.g., pair programming, prototyping). Evident from this rendering is how risk relates to individual requirements and which

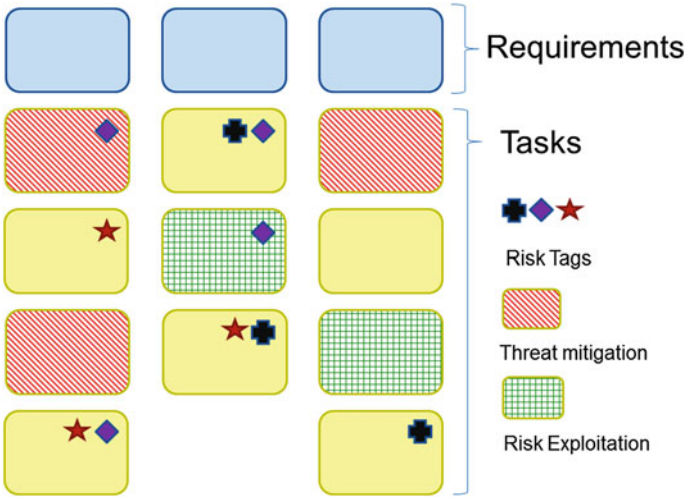

Fig. 8.9 Visualization of Distribution of Risk and Reward. Published with kind permission of © Alan Moran 2015

requirements are attracting the most risk attention. This permits someone not familiar with the technical details to gain an immediate insight into the risk profile of the iteration and to target their questions appropriately (e.g., lack of risk treatment of a requirement might suggest that risk identification was not adequately performed or the presence of a high degree of treatment may question whether the reward expected from the requirement is warranted in light of the risks).

It is in keeping with the transparency principle of agile risk management that all risk related artefacts (e.g., risk tailored agile chart, risk profile, risk log, risk burndown and risk-adjusted Team board) are all present and maintained in a common area visible to all project stakeholders.

8.8 Programme Risk Management

The DSDM Agile Programme Management Framework suggests that programme level risks be managed by the Programme Manager though of course multiple supporting participants may also be involved. Since a rather top-down approach is suggested (i.e., risks that appear to be linked to individual projects should not be included at the programme level), there is little guidance in respect of risk aggregation that might otherwise might apparent hidden risks of a systemic nature [129, 130]. Moreover, an implicitly negative frame is assumed and there is little mention of the risk related opportunities found elsewhere in the DSDM literature. Some of the common sources of risk (e.g., time and budget overruns, delivery of inappropriate capabilities, misunderstandings and poor communication between stakeholders)

are mitigated to an extent owing to the iterative and incremental approach of agile programme management as well as the focus on communication and collaboration. Several areas of concern for programme level risk include the continued ownership by the Programme Business Owner (which as can be seen from Chap. 5) is a crucial role at the programme level, the centrality of core artefacts (e.g., Business Architecture Model) and practices (e.g., project-level empowerment, adherence to Agile). There does not, however, appear to be a linkage to enterprise risk management in the sense that the introduction of new capabilities and the adaptation of business processes will inevitably reshape the risk landscape in which the organisation finds itself which in turn may create new threats or opportunities.

8.9 Management Implications

The fundamental goal of risk management is the balance of risk and reward wherein the pursuit of the latter rarely entails the lack of the former. There is little need to completely reinvent the risk management wheel for agile project, however, some amendments as outlined in this chapter do significantly improve both the acceptance and the effectiveness of risk practices in agile environments. Needless to say, from governance and financial standpoints the addition of risk management is welcome if it uncovers issues of process efficacy and efficient use of organisational resources. Furthermore, the linking of project and enterprise risk management helps alleviate an "island mentality" by connecting the issues that arise in projects to the wider programme and enterprise context. What remains of operational risk management (e.g., what/why technique, risk tagging and burndowns, risk walling) is merely a socio-technical translation into the agile arena of practices that have been found to be effective elsewhere. Indeed, validation of risk visibility is as simple as observing what is reported (e.g., risk-modified requirements map) and attempting to engage in meaningful discussions with the team. Whilst the specifics of the adoption of agile risk management vary as much by methodology as they do by organisation there already exists much guidance in the literature with which one can begin.

Cultural influences on risk management are particularly important in projects that employ outsourcing models that span multiple cultures. Setting aside for a moment the fact that an individual is the product of their environment and may be exposed to several different cultural influences, seen in aggregate the relative comfort that some exhibit towards risk (e.g., Chinese, Indian) may give rise to tensions with those from who show more uncertainty avoiding tendencies (e.g., Japanese, Korean, Russian). Combined with other cultural dimensions (e.g., respect for authority) there may even be an unwillingness in some cultures to draw attention to perceived risks if these are seen to be at odds with more senior managerial opinions. Added to this the pronounced tendency at a personal level to determine first how a risk is to be tackled and then afterwards rationalize this response in terms of risk exposure it becomes clear that management of the social and cultural influences requires astute and emotional intelligent leadership. The best advice here is to search for differences

in premises on which arguments are based rather than attempt to resolve their inherent rationality. The collectivism of agile risk management can both help and hinder in this respect owing to the diversity of opinion, though provided that there is a trusting environment wherein views are permitted to be challenged and an awareness of the issues on the part of management this ought to be resolvable.

Chapter 9
Configuration Management

Abstract Configuration management has sometimes been considered an ancillary discipline within the agile community in spite of its widespread use within the IT sector where it underpins practices such as continuous integration and automated deployment processes. Without configuration management it becomes impossible to effectively manage change, support exploratory feature development, ensure accountability during solution development or provide a holistic lifecycle view of solution development and agile process performance. Thus configuration management ought to be understood as much more than a mere technical means of tracking changes but rather as an important mechanism for supporting governance and risk management of agile processes.

9.1 Introduction

Configuration management [209] provides lifecycle management of entities, known as configuration items, that reflect the state of a project at any point in time. It is concerned not only with the entities themselves but perhaps more importantly the relations that exist between them. Generally associated with inventorising a project's assets, it is often considered to be a means of assessing the impact of changes and performing a tracking and control role within a project. Thus, by retaining an audit of all changes made along with a record of significant milestones, configuration management can provide deep management insights in matters of accountability and traceability (e.g., who changed what when), reproducibility (e.g., recreating a past state of the solution) and control (e.g., understanding of the impact of changes). Moreover, the relations between entities can reveal profound patterns relating to the solution and the process used to create it.

Irrespective of the nature of configuration management (or indeed if it has even been implemented), the case for programme configuration management is substantially stronger owing to the complexity of managing multiple on-going projects each sharing a common basis in terms of programme artefacts (e.g., Business Architecture Model, Programme Control Pack). Indeed, implementing and enforcing configuration management at this level can have considerable benefits from the perspective

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_9

of the Programme Team who must establish some form of oversight of the state of change at the programme level. Configuration management can therefore be established as a function within the Programme Support Office with input from the Programme Technical Co-ordinator and where appropriate Technical Co-ordinators at the project level to support the roll-out of configuration management across the projects. This necessitates the formulation of a Communication Management Strategy that should state baselining policy (e.g., Programme Foundations and Tranche Review conventions) as well as covering matters of configuration planning and audit.

9.2 Principles and Practices

Configuration management is concerned with the identification, recording and maintaining of information relating to the elements of a system, each of which is referred to as a configuration item (CI). Examples of CIs include software source code, build and project management artefacts. When applied over the entire lifecycle of the system, configuration management achieves a high degree of transparency and control over change. A Configuration Management System (CMS) comprises of the processes, practices and tools required to implement configuration management including a repository, usually known as a Configuration Management Database (CMDB),[1] wherein all configuration items along with their history and metadata are stored. Note that each configuration item is configured using a fixed pre-defined set of common attributes stored in the CMDB. A CMS will typically interface with its CMDB through an appropriate form of tooling so that direct access to the CMDB and its assets are not permitted. Furthermore, a CMS may impose access rights controls (or derive these from a third party security provisioning system) in order to control who may access and alter configuration items and also ensures that the historical records of selected assets cannot be altered in the future (e.g., baselined versions of documents or previously deployed solutions) as well as assisting in day-to-day activities (e.g., retrieving copies of past versions of configuration items). Technical matters aside, configuration management is an important process that supports governance, risk management and evidence-based decision-making. Achieving the requisite level of maturity remains, however, a challenge that requires discipline albeit one with significant rewards.

Classical configuration management traditionally adheres to the following pattern of activities which are presented here in a more nuanced form appropriate for agile environments:

[1] The configuration management database is merely a datastore and need not be a relational database. Often such databases are in fact versioned directory structures in filesystems.

- *Configuration Management Planning*. This involves the high level organisation of configuration management and usually takes the form of corporate standards adapted to the project circumstances (e.g., choice of configuration management tools, naming conventions). Agile teams often use this phase to establish initial naming conventions and working practices (e.g., the relationship between configuration management and continuous integration).
- *Configuration Identification*. Perhaps the most crucial aspect to configuration management is the determination of the granularity of configuration items along with their attributes. Once defined this enables lifecycle management of items. Particularly within agile IT teams there already exists a broad consensus as to what constitutes the appropriate scope and depth of configuration identification (e.g., versioning of software code, handling of project artefacts as individual CIs).
- *Configuration Change Management*. Controlling changes to configuration items traditionally involves approval processes as well as controlling the manner in which modifications are applied. In agile environments where change is acknowledged and embraced it is important to ensure high levels of automation (e.g., authorisation checks) in order not to impede the solution development process.
- *Configuration Status Accounting*. This is where configuration management really proves its worth by ensuring high degress of transparency and accountability. Since configuration control plays a lesser role in agile environments, the primary function of status accounting is to be able trace the effects of changes in order to be able to respond appropriately to changing circumstances rather than to support audit approval decisions. Audit functions of a CMS can, however, assist when attempting to ascertain the origin and impact of changes. Typically in agile environments these contribute to learning by providing highly automated and effective feedback loops (e.g., the use of configuration management together with continuous integration practices to identify at what point in time a deliverable ceased to be functional).
- *Configuration Verification and Audit*. This is concerned with the integrity, historical accuracy and reproducibility and traceability of assets. Thus is may be seen in terms of quality assurance of the configuration management process in so far as it encompasses verification of the process and auditing of configuration items as well as compliance to policies and standards. In an agile environment the exercise of comparing a process implementation with a pre-defined specification carries less credibility than a culture of reflective adaption of the process to meeting changing needs. Accordingly this facet of configuration management may be subject to considerable reinterpretation along agile lines (e.g., assessment of CI attributes in order to assess the levels of compliance with working agreements). In highly automated environments this is often embedded within the practices and tooling of a CMS (e.g., compliance may be enforced by commit level controls such as the automatic prohibition of commits without supplementary comments explaining why a change has been made). An over reliance on tooling, however,

Table 9.1 Configuration Items

Entity	Description	Rationale
Project Artifacts	All products relating to the management of the project (e.g., Business Case, Prioritised Requirements List, Development Approach Definition)	All such artifacts are usually baselined at the end of foundations which implies a capturing of the state of the project at that point in time
Evolving Solution	All elements that comprise the evolving solution (e.g., Models, Prototype, Tests, Components)	Being in a position to manage change requires a high degree of control over it (e.g., comparison of different states of the solution)
Deployed Solution	Each increment including its packaging with other artifacts necessary for its operation	It is commonplace for a deployed solution to migrate its way to the production environment whilst undergoing final testing. It is therefore important that the integrity of the solution remains intact. Furthermore baselined solutions make possible rollback scenarios which need precise information concerning past states of the solution

can prove naive and misguided thus necessitating some form of review (e.g., a visual inspection of how controls are been complied with). Furthermore, an agile perspective on configuration management and auditing necessarily implies that adequate reflection on how a CMS is being deployed[2] must take place and that necessary improvements be implemented on a regular basis during the project (e.g., during the Review at the end of each Timebox).

It is important to ensure that the team and its stakeholders understand the purpose and value of configuration management and that the breadth of configuration management activities is determined e.g., the integration of configuration management into the wider configuration management used by an organisation. Despite the optimistic tone of the configuration management literature, however, many firms struggle to implement it at the enterprise level, an undertaking take can take months to years to complete. The scope of a CMS should ordinarily be understood to include everything within the project remit (e.g., models, prototypes, sources, deliverables, test artefacts) and Table 9.1 gives some indication of what this might entail on a DSDM project though precisely how this is structured will of course vary (e.g., reference may be made to enterprise coding standards maintained at a higher level in the organisation). From an organisational perspective, DSDM associates championship of configuration management with the Technical Co-ordinator who must therefore

[2] A solution development team may decide, for example, to perform additional commit level checks to assess consistency and quality of changes.

play a key role in planning (e.g., proposal of standards and conventions). The role of Configuration Manager itself (if it is even required) could thus be subsumed under the Technical Co-ordinator or perhaps either the Project Manager or Team Leader as appropriate.

Considered as configuration items, the output of metrics (e.g., test coverage and static analysis reports) provides a powerful reporting mechanism that enables a holistic assessment of the solution and the process used to create it. Combining these with existing configuration items such solution and related artefacts (e.g., deliverables, PRL, Timebox Review Records) or incident reports, this builds a more complete picture with which evidence-based decisions can be made. For example, does increased test coverage contribute to fewer incidents and at which point do such efforts become marginal? Which solution quality attributes contribute most towards architectural integrity and how have these evolved over time? Is the scale of change (e.g., numbers of commits between baselines) contributing to solution instability or impairing overall quality? Indeed, it is the search for the answer to such questions that shapes the design and content of a CMDB.

Configuration management is, however, one of the more difficult processes to establish and maintain. It requires considerable planning and effort to record configuration items and discipline to comply with the process. This seemingly burdensome overhead does at first glance appear to suggest that it could impede both the practices and culture of agile environments. Yet, configuration management appears, at least partially, to be in widespread use in most agile projects in the form of Software Configuration Management (SCM), the most commonly cited example and a key enabler of core practices (e.g., continuous integration, refactoring). This suggests that two key success factors driving the adoption of configuration management are the determination of the appropriate level of granularity of configuration items and the level of tool automation to support the process. Indeed, a common mistake when starting out is to define configuration items at too fine a level of granularity causing the effort to record and maintain changes in a configuration management database to spiral out of control later. The litmus test in such situations is to ask what are the questions that must be answered for which that level of configuration detail is necessary and whether or not the team is prepared to maintain those attributes in order to ensure that such questions can always be answered. Thus since changes require continually updates to configuration item attributes the level of automation clearly plays a decisive role. In fact, it is precisely the wide availability of configuration management tools that makes (software) configuration management so ubiquitous and successful in the IT sector.

One specific example of configuration management in IT projects is the widespread use of software configuration management (SCM) usually implemented as version control systems. This enables every change to the code to be recorded (e.g., who made the change, when was it made) often with additional information (e.g., comments made by the Solution Developer when committing changes). In this context the configuration item (CI) is the software under development though this is usually framed in version control language. For example, a software product that comprises of a core component along with a graphical user interface module may be either be classified as a single CI or two separate CIs.[3] The classification of configuration item is generally kept independent of the packaging of the solution. For example, in software development it is common to gather all related components of a solution together and deliver these as a single artefact. This notwithstanding, it is still possible to place deliverables under configuration management provided these can be linked back to the code from which they were built. Benefits, often reaped on a daily basis, include the ability to trace the precise changes that gave rise to defects and the provision of timely feedback information (e.g., email notifications via continuous integration systems of code versions that are deemed to have issues). For example, search algorithms in some version control systems (e.g., mercurial, git) are very advanced and can even be linked to tests i.e., to find the point in the version history where a specified test first started to fail [154, 199].

The operational aspects of configuration management are illustrated in Fig. 9.1. Thus, an individual wishing to make a change to a configuration item must therefore acquire a working copy of this repository using a one-off operation that is referred to as a check-out. Thereafter changes may be applied and committed back to the CMDB (at which point new versions of the respective items are created) and updates from other authors can be acquired and integrated into the local copy. Since there is a potential for conflict arising from two authors making different changes to the same item, configuration management systems are enabled with sophisticated conflict resolution mechanisms that resolve most technical issues. This notwithstanding some conflicts do require communication between the respective authors and therefore the immediate feedback that regular commits and updates offer is enhanced by a culture of communication and collaboration between individuals. Moreover, access to historical records of change along with automated mechanisms for finding significant points in time considerably enhance the value of a CMS in managing change.

Over time different configuration items will evolve at different rates depending on the number of committed changes and hence versions. There are, however,

[3] Distinct CIs may be nesscessary if the core component and the graphical interface are in separate version control locations and would certainly be sensible if either can ever be used independently of the other.

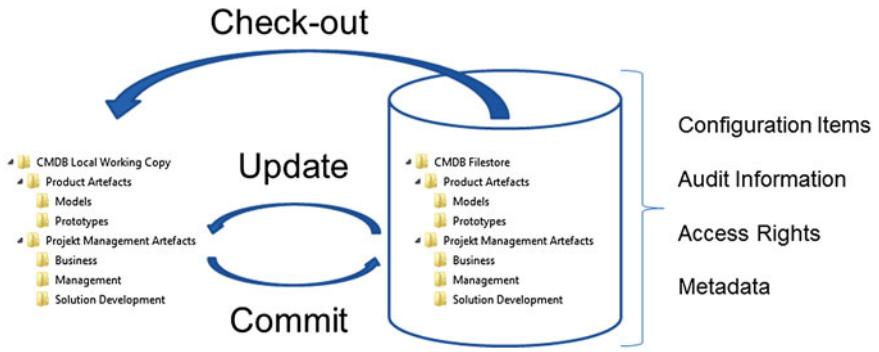

Fig. 9.1 Configuration Management Operational Model. Published with kind permission of © Alan Moran 2015

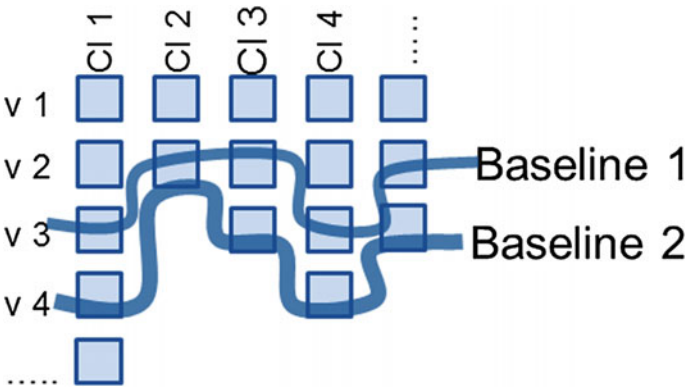

Fig. 9.2 Baselining Assets in the Configuration Management Database. Published with kind permission of © Alan Moran 2015

milestones that require capture of the state of the entire project at specific points in time (e.g., conclusion of the Foundations phase). This is known a baselining[4] and is illustrated in Fig. 9.2 which shows one baseline containing version three of the first configuration item, version two of the second and third configuration item and version three of the fourth followed by a second baseline that contains version four of the first configuration item, version two, three and four of the second, third and fourth items respectively. Thus during the period between these two baselines the second configuration item was subject to no change whereas all other cited documents were changed once. Such baselines enable the recreation of the entire project at specific points of time in the past (e.g., a check-out of all documents as they stood at the end of the Foundations phase) and help reconstruct and understand events of the past (e.g., comparison of documents between the Feasibility Assessments and Foundations Summary milestones). Frequent baselining is advocated in DSDM (e.g.,

[4] Some configuration management systems use the term tagging to refer to the creation of a baseline.

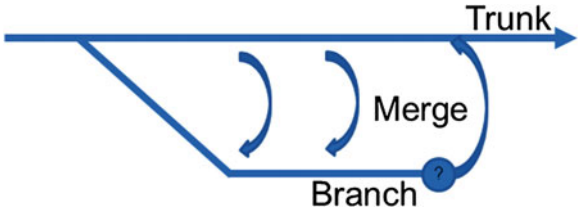

Fig. 9.3 Simple Branching Model. Published with kind permission of © Alan Moran 2015

as part of continual integration, around key events such as demonstrations or decisions, at transition points between Timeboxes) which entails attributing metadata to all configuration items present at that point in time. Indeed, the practice of including all existing configuration items in a baseline has the advantage of making possible detection of items that were absent at that point in time but added later.

Another capability that arises in configuration management is the ability to define branch points that facilitate explorative activities as depicted in Fig. 9.3. Here the efforts of the team that are focused on ongoing solution development are represented by the trunk whereas exploration (e.g., modelling, prototyping) that requires potentially damaging or disruptive alterations to the trunk are isolated on a separate branch. During the exploratory periods continual merges of changes arising from ongoing work on the trunk are integrated into the branch so that changes are synchronised in the event that the branch should later rejoin the trunk (by which time the necessary adaptations on the branch have already been incorporated). On conclusion of branch activities and in light of the learning that has taken place a decision has to be made to either merge the branch back into the trunk or discard the work entirely. Without the support of a configuration management system such undertakings would present an unnecessarily high risk to the project owing to the conflicting needs of trunk and branch and the inability to isolate and resolve them. Though branching is a widely used technique, it can give rise to subtle issues when multiple branches are in operation in parallel though these are inherently solvable. One issue, known as a semantic conflict, concerns changes in the interpretation of solution features on one branch where its underlying implementation has changed in another as described in [184].

Owing to the contribution that configuration management makes to the visibility of a solution configuration (i.e., what components comprise the solution, their functions and attributes) it is often associated with other practices (e.g., quality management, inventory management, auditing) though these disciplines are distinct and serve different purposes. For example, the linkage of solution components with their test plans and incidents that arise during their operational use can make clear how effectively testing strategy (i.e., quality control) is actually contributing to solution quality and whether process improvements are necessary (i.e., quality assurance).

9.3 Agile Metrics

A significant feature of evidence-based approaches is the capability to report progress in terms of well-defined metrics at the change set level. The agile manifesto asserts that working solutions are the primary measure of progress and that regular collective reflection paves the path to process improvement. Agile metrics play a central role in this assessment though several of the measures that feature prominently in agile projects (including those listed in Table 9.2) focus on process performance without necessarily providing value based insights. Alternative approaches, linked to the Scum framework, endeavour to measure value, diagnose capabilities and improve continuously as the basis of an agile interpretation of evidence-based management [232] that represents a shift towards the analytics of outcomes in terms of an analysis of investments and initiatives. For example, some common measures focus on:

- *Current Value*. This includes revenue per employee, product cost ratio (i.e., an attempt to capture product related expenses) and varies measures of satisfaction.
- *Time-to-Market*. Indicators such as release frequency and stability and use of cycle time metrics (i.e., the time required to satisfy customers or respond to market opportunities).
- *(In)ability to Innovate*. These include installed version index which tries to assess the difficulty that customers face when installing new releases of products, usage index which measures how difficult a product is (including the burden of sustaining rarely used features), innovation rate (a measure of technical debt and cumulative costs of sustaining such solutions), defect rates.

These can be used to baseline current performance as the basis for incremental improvement based on practical experiments to improve value.

Metric-driven approaches are doubtless familiar to those already acclimatized to process based environments (e.g., ITSM, COBIT®) where quality improvement practices feature strongly (e.g., PDCA [65], Kaizen [127]) and their limitations in driving behaviours are equally well acknowledged. This recognition is based on the inherent difficulty in determining appropriate individual measures that can be used as the basis for key performance indicators though this is offset somewhat in agile environments that consider metrics as a feedback instrument to inform organisational learning. When analysed in terms of their dynamic and relationship with each other rather than individually, new insights can, however, be gleamed. This is where configuration managment systems and their related data-driven analytics come into play. Indeed, many innovatives enterprises attribute their success to the manner in which they can integrate product development with lifecycle analytics [227].

Table 9.2 Sample Agile Metrics

Metric	Description	Comments
Velocity	The sum of estimated effort of all completed user stories within a specified timeframe	Velocity is usually measured in terms of user story points at the Timebox level. Partially completed stories do not contribute towards velocity
Test Coverage	The percentage of the solution implementation that is covered by test cases	This metric is commonly used in software development environments and can take on varying expressions (e.g., coverage by line of code or by module). High coverage ought to ensure high rates of defect detection and is particularly valued when used in conjunction with practices such as test-driven development, continuous integration and refactoring
Static Code Analysis	Class of metrics that are apply to the inspection of the implementation of a solution	This metric is commonly used in software development environments and requires access to source code though execution thereof is not required. Typically indicators of complexity, nature of data flows and model correctness are the focus of such metrics which often identified issues to be addressed at an architectural level (e.g., circular dependencies)
Burndown	Plot of quantity of work remaining against time	Burndown charts are often rendered at the Timebox, Iteration or Project levels that estimates velocity. Actual burndown is usually plotted against a nominal linear planned burndown
Lean Metrics	Class of metrics that focus on flow during a prescribed timeframe	Typical measures range from how long it takes for a feature to be developed to the point of acceptance testing (i.e., feature cycle time), extent of customer satisfaction (e.g., net promoter score) or overall efficiency (e.g., process cycle efficiency)
Team Motivation	Reflective collective metrics indicating how motivation was impacted by events or practices	Creative examples include timelines citing key events that influenced motivation (e.g., happiness line) or scorecard based rating systems that collate satisfaction levels with existing practices (e.g., rankings of practices using a five point Likert scale at the end of each Timebox)

9.4 Management Implications

Configuration management provides considerable benefits in terms of governance, risk management and effective decision-making. Indeed, the intrinsic nature of a CMS has the potential to define to a very large extent the manner in which a team is expected to work whilst also providing immediate transparency in terms of compliance with established working practices. This potential rests on the extent to which CMS practices can be automated and integrated with interfacing systems that are external to the project (e.g., incident reporting systems) though care should be taken not to introduce processes and tooling at the expenses of individuals and their interactions. Thus the artefacts on which a project team works can, for example, be placed under version control thereby providing a non-invasive avenue into configuration management that ought to work reasonably well provided that all team members are sufficiently conversant with working in this manner (e.g., resolving change conflicts). Furthermore, team members may find it beneficial to determine a set of working agreements that can be codified and enforced by the CMS though not every convention of the workplace needs to be handled in this manner. The issues tend to arise when the team has over estimated its own ability to adhere to its ideals which in turn requires a flexible and adaptive approach to how the CMS is to be deployed. If such infrastructure is maintained by centralised specialists who are not available on demand, then this situation can impede learning. Moreover, if interfaces to external systems are involved (e.g., the integration of an incident reporting system against which quality and testing attributes are correlated) then these too can introduce delays or complexities that impair decision-making. The general advice, therefore, in regard to configuration management is to start small and incrementally build up capabilities in line with management needs to address specific issues the project is facing (e.g., tracking of effectiveness of quality over the entire lifecycle of a solution).

Configuration management can also demonstratively reduce risk encountered in projects by enabling the Solution Development Team to engage in changes without fear of losing control or oversight. For example, faced with the considerable uncertainty that the introduction of a major change is likely to have on the current state of a solution, a team might elect to branch in order to learn and gain deeper insights into the feasibility and validity of a proposed solution. This can be done in a collaborative manner delaying the integration decision until such point in time as the proposal has been validated. In this context there is little to no uncertainty surrounding undesirable side effects, the strategy to take if the team decides not to go forward with the change or the likely impact it will have if it is adopted.

Perhaps one of the most promising, though arguably most difficult to attain, aspects of configuration management is its capacity to inform lifecycle and process decision-making. The relationships that arise between solution artefacts, their quality and testing attributes and metrics relating to their operational performance provide profound insights into the solution and the process used to create it. This can yield valuable information to the organisation that may be applied to other projects (e.g.,

which quality metrics are most effective at predicting future performance). It can also proactively track and monitor dynamics within the project that may require management attention. For example, recall from the systems dynamics analysis of agile processes (see Chap. 1) that there is a relationship between schedule pressure and implementation churn and their impact on the extent to which test-driven development and refactoring are practiced. Analysis of this situation can give indications of rising technical debt and depletion of solution quality which must be addressed before the trigger point of a major refactoring is reached. This example also illustrates how reliance on CMS analytics alone will never suffice since the same systems dynamics indicate that pair programming, the practice of which is not recorded in the CMS, is instrumental is resolving this scenario. The key point to focus on is that such analytics serve to enhance learning within the team and help direct them towards desirable outcomes. Accordingly, the design and content of a CMS should primarily reflect the managerial and operational needs facing the project and be capable of adaptation when these change. As a final note it is often unhelpful to link such information systems to performance appraisal methods since the rigidity and divergent goals of the latter have the tendency to warp the good intentions of the former.

Part III
Organisation and People

Fundamental to the operation of Agile is the nature of an organisation and the people found therein, both strands being intrinsically linked and bound to the fortunes of the enterprise. At the organisational level culture has a profound influence on the success of Agile which thrives in an atmosphere of openness and continual learning. Therein lies the central paradigm of self-organisation replete with elements of adaptibility and flexibility, reflected in the pyschology of individual employees bound together by the motivation, commitment and emotional intelligence that yields high performance teams.

Chapter 10
Organisation

Abstract Organisational theory embodies a wide range of concerns, though its primary focus is on the operating characteristics and structure of the enterprise. On account of its prevailing influence, it is important first to understand organisational culture and how this impinges on and is influenced by the adoption of Agile practices. This naturally leads to a discussion of organisational learning owing to the intimate links learning shares with culture and how these combine to ensure that outcomes are achieved in an agile organisation that adopts an adaptive future looking and change driven orientation. What emerges is that self-organisation becomes the central organisational paradigm, endowed with the necessary empowerment and autonomy required by agile teams willing to assume initiative and responsibility. This in turn prompts a different leadership response from management that is more aligned with these underlying cultural and learning drivers. Finally, attention turns to scaling Agile and the impact this has on those key ingredients that have made Agile so successful at the project level raising the question of whether or not they can be sustained in larger change programmes.

10.1 Introduction

One imagines that organisational theory is primarily concerned with the structures (e.g., organisational charts and roles), responsibilities (e.g., job descriptions) and relations (e.g., between staff and management) that are bound together within the culture that defines how work is conducted within the enterprise. In reality the workplace is a fusion of working, learning and innovation all of which are forces that have conventionally been considered to be at odds with each other [35]. Drawing these elements together is the realisation that abstracted learning ought not to be detached from working practice through which it becomes validated and refined as result of innovation. For agile environments learning is central and for this to flourish learning must become incorporated into the organisational culture along with adequate supporting infrastructure and leadership from the top (e.g., knowledge vision). This

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_10

enables the organisation as a whole to better marshal its intellectual assets and pursue sense-and-adapt strategies towards the external environment. The union of working, learning and innovation also requires flexible and fluid organisational structures in which groups are permitted to form with sufficient autonomy in order to question established ideas and develop new working practices.

In promoting flatter hierarchies and more fluid structures, agile organisations admit the possibility of open and viable communities of practice [275] wherein members are bound together by a common sense of purpose (rather than shared skills or profession) the growth of which relies on the sharing of information and experience. This group identity fosters a sense of membership defined in terms of expertise or tacit knowledge which may be context based and difficult to codify or store externally [61]. Thus any discussion of organisation must cover organisational culture and learning and examine what organisational paradigm is most appropriate for purposes of Agile. If this is to have wider relevance beyond the project context then the question must also be asked if the practice of Agile itself can be scaled up to the wider organisation.

10.2 Organisational Culture

Organisations exist within a web of customers, suppliers, partners, competitors and other stakeholders (e.g., regulatory authorities, general public) and its behaviour therein assumes the character of those patterns of action and understandings that constitute its culture [217]. Organisational culture is usually defined in terms of the shared assumptions learned in the course of problem solving and deemed to be have sufficient general validity to be passed on to new members [228]. This focus on culture suggests that it is something learned and essential to the manner in which work is conducted within the organisation. However, organisational culture can also embody belief systems which endows an organisation with meaning [62] and establish collective mental models to which individuals from different backgrounds subscribe [117].

Whilst there may be a prevailing view with regard to what constitutes the dominant culture within an organisation, this does not imply that that culture is uniformly defined, agreed upon or even accepted as evidenced by the existence of subcultures within an organisation [175]. The extent to which an organisation can truly become agile is nonetheless believed to be primarily linked to its underlying dominant corporate culture. In particular it would appear that more democratic organisations have a higher likelihood of success with Agile owing to their horizontal hierarchies, flexibility and spontaneity [242, 259]. In contrast plan-driven approaches, when matched with policies and procedures, are more appropriate for stable environments that require well-defined roles and tasks [32].

Organisational culture is manifest at three levels (see Fig. 10.1 and Table 10.1) reflecting the extent to which it is visible to the observer [228]. These levels highlight the pitfalls of assessing Agile in purely superficial terms owing to the notorious

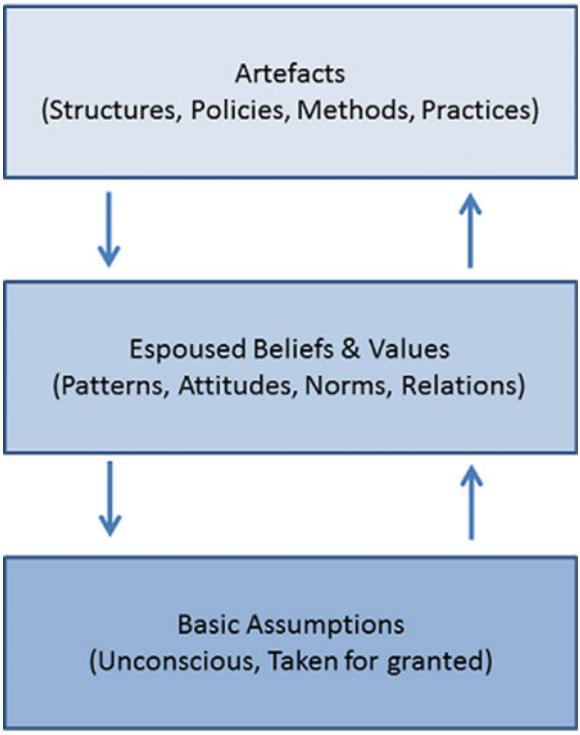

Fig. 10.1 Three Levels of Organisational Culture. Adapted from [228] Published with kind permission of © Alan Moran 2015

difficulty of deciphering intent[1] because of the tendency to project an interpretation based on one's own past experience. For example, the atmosphere and physicality of an agile environment might appear chaotic to someone more accustomed to phased planning approaches where the absence of specific artefacts might be misinterpreted as an oversight. Attached to this are cultural artefacts such as symbols, rituals and routines[2] evident in agile communities in the form of events (e.g., stand-up meetings, collective celebration of delivery of a solution increment), objects of significance (e.g., Prioritised Requirements List, agile chart) or common methodological language all of which must be absorbed and comprehended in order to be able to adequately measure up the situation. In such circumstances experience within the group along with exposure to deeper (i.e., less visible) levels of culture would be beneficial. This perhaps explains to an extent the attitude of some stakeholders who despite being members of an agile organisation have not yet grasped its significance

[1] The example of Egyptian hieroglyphs and symbolism which, whilst highly visual is also profoundly difficult to understand, is cited as an example of the difficulty of cultural analysis when limited to artefacts.

[2] Symbols are objects, words or actions to which a particular significant or meaning has been attributed, rituals are collective activities that are considered socially essential and routines are regular sequences of action that have become well-established and accepted.

Table 10.1 Three Levels of Organisational Culture

Level	Description	Agile examples
Artefacts	All visible, auditory and tangible phenomena of the culture including physical environment, language, technology, presentation (e.g., clothing, style), mannerisms, processes, documents, observable rituals and ceremonies	Agile team (as empowered, motivated and self-organised), events (e.g., stand-up meetings, adaptive planning), techniques (e.g., pair programming, test driven development, refactoring, MoSCoW prioritisation, workshop facilitation), technologies (e.g., continuous integration, daily build, deployment pipeline), practices (e.g., iterative development, incremental delivery, customer integration) and documentation (e.g., agile charting, risk walling, burndown charts, prioritized requirements list)
Espoused Beliefs and Values	Shared understandings that grow from group learning. Though some beliefs and values will later transform into basic assumptions, many will not as they are neither entirely reliable nor testable	The preferences and principles enshrined in the agile manifesto, self-organisation, inspection and adaptation, reflection and continual improvement, feedback integration, trust, openness, communication, collaboration, continuous learning and technical excellence
Basic Assumptions	Those beliefs and understandings that over time have been reliably validated through collective action and deemed to have general validity at which point they become taken for granted	Focus on business need, significance of personal responsibility, strength in diversity, importance of communication and collaboration, positive attitude towards change and uncertainty

Adapted from [259]

owing to their (experiential) distance from agile culture. It also makes clear that organisational culture can be a serious impediment to agile adoption since it involves not only changes to daily practices and rituals, but also realignment of beliefs and values that may ultimately challenge underling basic assumptions. This explains why so many elements of agile (e.g., customer involvement, assuming personal responsibility, coping with uncertainty) present significant challenges to practitioners and stakeholders alike.

One interesting classification of organisational cultures uses the Uncertainty Avoidance (already encountered in Chap. 8) and Power Distance dimensions, that are based on well-established culture research, to derive a matrix of four possible types of culture [118, 242] as shown in Fig. 10.2. Uncertainty Avoidance expresses the degree of discomfort felt towards uncertainty and ambiguity whereas Power Distance captures the extent to which those with less influence accept and expect that power is distributed unequally.

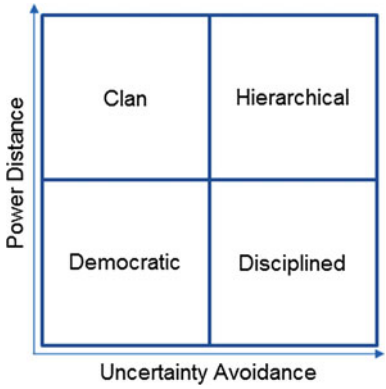

Fig. 10.2 Typology of Cultural Types Based on Cultural Dimensions. Adapted from [242] Published with kind permission of © Alan Moran 2015

- *Clan*. This describes a traditional, but loose culture whose leaders are mentors and facilitators. Bound together by loyalty and tradition, there is a pervading sense of cooperation and security towards participant in which cohesion and particular characteristics of personal expression and conduct are valued.
- *Hierarchical*. Pronounced hierarchies with strong leadership and clear assignment of responsibilities are the hallmarks of this culture type wherein respect for status and a large degree of formalism can be found. A product focus can often be found in such organisations.
- *Democratic*. This culture admits a high degree of flexibility and spontaneity where initiative and personal responsibility are valued. Leadership revolves around coordination and organisation based on flexible rules and negotiated problem-solving. There is a strong people orientation in which individuals are expected to contribute to the overall growth and development of the organisation.
- *Disciplined*. Similar to hierarchical organisations, there is an emphasis placed on discipline and formalism with participants expected to demonstrate self-control. There is a strong project orientation with importance attached to productivity and efficiency.

Clearly a Democratic cultural type is most conducive to agile environments since it is likely that the necessary empowerment can be easily granted with which collaborative participation and consensus seeking can be directed towards problem-solving. This does not preclude agile activity occurring in other types of cultural environments, though it does make clearer what challenges may lie ahead (e.g., the stifling effect of formalism and hierarchy when trying to induce members of a team to act with initiative and personal responsibility). There is no clear unequivocal answer concerning how best to tackle such cultural compatibility since the agile approach has in fact been found to thrive in very diverse types of organisations [217]. Added to this is the fact that there is evidence of interplay between organisational culture and the application of agile methods which suggests that there is room for marriage in either culture change or the adaptation of Agile.

Internal cultural aspects aside, however, the nature of the environment in which any methodology finds expression also plays an important role. Categorised in terms of the extent of change (i.e., stable or volatile) and the nature of focus (i.e., internal or external), it is clear that it is in high change, externally focused environments that agile methodologies are best placed owing to their natural embracement of change. Since environmental circumstances define the terms of reference for activities, this analysis helps better an understanding of to what extent cultural adaptation is called for. For example, in stable and predictable environments the power distance of Clan and Hierarchical culture types is not as serious as it might be in volatile environments where timeliness and accuracy of decision-making is crucial. Equally stable and predictable environments exhibit less of the vulnerability towards uncertainty found in Hierarchical and Disciplined cultures whose behaviours and modes of operations are therefore considered less inhibiting. Therefore it is the match between the dominant organisational culture and the relationship the organisation bears to its environments that is the key determinant in deploying or adapting agile methods to ensure their success.

Many of the reported cultural characteristics of agile organisations include not only the democratization and flexibility alluded to above and traits such as openness, tolerance and trust referred to elsewhere, but also a positive attitude towards learning and working with others in teams. Since culture is rooted in group learning, this requires organisational support (e.g., regular reaffirmation of core values) and resourcing (e.g., learning infrastructure) in order to build and sustain it over the long term.

10.3 Organisational Learning

If organisational learning is understood as a process of planning, execution, reflection and adaptation then group learning must be based on activity, experimentation and reflection that permits individuals to question underlying assumptions and openly discuss differences of opinion. Framed in the expectation of the efficiency of routines based on an environment that experiences little change, experimentation is often considered a waste of time. In high volatility environments, however, the potential gains are significant (and therefore the associated risks of experimentation are low) as all stakeholders endeavour to gain a common understanding with which to determine a new path to value.

The genesis of many of the ideas concerning the exchange of information in agile environments can be traced back to notions of knowledge creation and management [256]. In this context tacit knowledge (i.e., subjective and experience based understandings that are often context specific and reflect beliefs and intuitive models) is contrasted with explicit knowledge (i.e., objective, rational and very often context free). Knowledge is distinct from information in the sense that it reflects underlying beliefs, is related to specific actions and bears meaning in relational terms. Its creation may be classified in terms of the extent to which it is expressed in tacit or

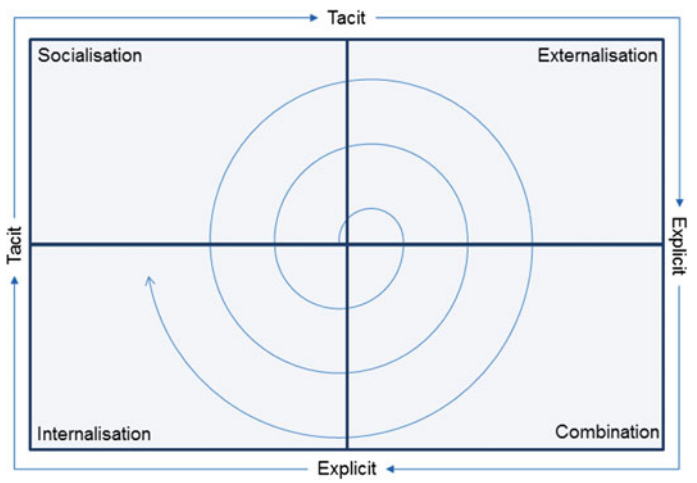

Fig. 10.3 Tacit and Explicit Knowledge and the Four Modes of Knowledge Conversion. Adapted from [256] Published with kind permission of © Alan Moran 2015

explicit forms (i.e., epistemological) and where it resides within the organisation in terms of individual, group, organisation or inter-organisation possession (i.e., ontological). Conversion of knowledge[3] is a continual and spiral movement between these two types of knowledge that involves one of four possible modes as illustrated in Fig. 10.3:

- *Socialisation*. This involves a tacit-to-tacit transfer knowledge wherein direct experience between individuals plays a significant role. Owing to their integrated and heterogeneous nature, this form of transfer is commonplace in agile teams. This also explains why co-location (or appropriate proxies such as remote conferencing) is so important in agile environments. The outcome of this type of exchange is that mental models and technical skills are shared. Pair programming is an excellent example of socialised learning.
- *Externalisation*. This concerns the making explicit of tacit knowledge and requires articulation and translation in order to find a common language and expression of ideas through which shared beliefs and differences are discovered. Often conceptual knowledge is derived during this mode and is a group undertaking that often benefits from a facilitated workshop setting. Modelling and prototyping are also good agile expressions of externalisation.
- *Combination*. The collection, processing and dissemination of knowledge is an explicit-to-explicit conversion process that ensures that knowledge can be shared more widely. Thus the result is the derivation of systematic knowledge that can be used in scaled Agile or wider organisational environments.

[3] Spiral learning within an organisation in terms of the four modes of knowledge conversion requires conscious initiation and sustaining of efforts.

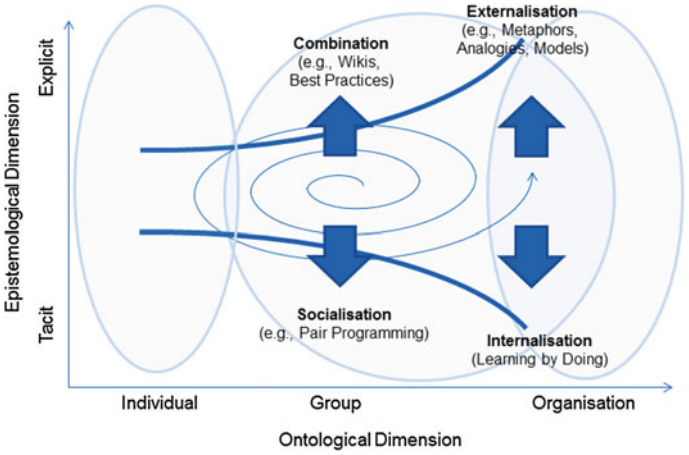

Fig. 10.4 Spiral Learning within the Agile Organisation. Adapted from [256] Published with kind permission of © Alan Moran 2015

- *Internalisation*. The absorption of concepts, models and methods such that they can be applied in a specific context is the result of an explicit-to-tacit conversion which enables individuals to interpret and apply knowledge established within the group context. This amounts to operational knowledge.

Thus spiral learning is concerned with the notion that knowledge grows through the interaction of the individual with others and that when appropriately managed ultimately becomes organisational knowledge. For example, as illustrated in Fig. 10.4, in agile teams techniques such as pair programming, architectural spiking and brainstorming activities illustrate socio-technical practices that promote learning (i.e., Socialisation). Arising from this is the articulation of experience and tacit knowledge into explicit concepts which is supported by collective reflection (e.g., facilitated workshops, retrospectives) using the devices of metaphors, analogies and models (i.e., Externalisation). In this context metaphors capture an intuitive impression of new ideas in symbolic terms that help reconcile discrepancies in meaning. Metaphors can be augmented by analogies that highlight differences based on rationalistic thinking thereby helping to establish a bridge between an image (i.e., metaphor) and a model which requires formulation of more precise language and concepts. This progression of thinking can be observed in agile modelling and related practices as ideas find implementable interpretations. Once new knowledge has been created it must be assimilated into the wider body of knowledge in order to present a coherent picture of the whole (i.e., Combination). This requires sufficient validation of the key concepts as well as their integration into working practices and artefacts (e.g., knowledge bases, promotion of best practices). This in turn impacts on how work is done which highlights once again the link between organisational learning and culture. Often Combination requires management support in terms of coordination and organisation. Finally this body of knowledge must find application in specific situations in the workplace on the basis of learning by doing. For this to happen,

there must be an undisputed basis for the existing knowledge (e.g., diagrams, documents, consistent verbal accounts). Internalisation also requires that individuals are prepared to experiment in their application of knowledge which in turn benefits from exposure to thinking outside of their ordinary domain of expertise. Heterogeneity, a key characteristic of agile teams, therefore supports these endeavours by widening the collective experience base within the group.

The value of a vision in assisting the articulation of shared beliefs should therefore not be underestimated since it acts as an initiator towards spiral learning within a group that enables the conversion of tacit knowledge to explicit knowledge (i.e., Externalization) that ultimately provides the basis for actionable effort. It is therefore not surprising that vision statements feature so prominently in agile methodologies.[4] Given the importance of spiral learning and its impact on agile thinking, it is evident that embedded learning is crucial for highly change oriented and innovative enterprises which profit considerably from structuring their projects along agile principles.

Therefore, if the purpose of a project is to translate an idea (along with its assumptions and underlying belief systems) into a working solution, then it follows that tacit knowledge and understanding must be articulated, translated, disseminated and acted upon. The early stages of this process involve Socialization and Externalization and benefit tremendously from self-organisation and agile practices. The later stages, however, require Combination and Internalization that often profit from division of labour and specialization. Thus what emerges is a hybrid enterprise that embodies elements of both fractal and bureaucratic organisational structures with the flexibility, adaptability and collaborative nature of the former compensating for the weaknesses whilst complimenting the efficiency of the latter [256, Chap. 6]. Enterprises that achieve this balance and have the management capabilities to cope with them are not only ideally suited to integrate agile practices, but can successfully scale them throughout the organisation. This perspective also explains why, contrary to popular belief, it is perfectly feasible to combine Agile with elements of more control oriented processes (e.g., CMMI, COBIT®, ITSM) providing that adequate mutual accommodation is present [249]. The role of management in this context is to initiate and support spiral learning and to act as a bridge between vision and reality. Thus organisational learning, insofar as it is involved in the creation of knowledge, requires the full participation involving of the following roles [256, Chap. 5]:

- *Knowledge Practitioners*. These individuals are intimately engaged in knowledge on a daily basis assimilating information from both inside and outside the organisation and can be classified in terms of the following two subgroups based on their command of tacit and explicit knowledge respectively:
  - *Knowledge Operators*. Based on the front line of business or technology, operators capture essential tacit knowledge gleamed through contact with others (e.g., customers, stakeholders). This enables discerning insights into what really merits attention during solution development. Example roles may include sales

[4] Examples of vision statements include the "System Metaphor" of Extreme Programming or the "Product Vision" of Scrum. DSDM subsumes visionary statements in its Business Case.

representatives, service desk or other skilled workers involved in operational aspects of the business.
  - *Knowledge Specialists*. This group is capable of mobilising well-structured explicit knowledge and dealing with it in conceptual and analytical terms. This category may contain research scientists, software engineers and specialists from a diverse range of fields (e.g., HR, legal, finance).

  Typically in an agile setting practitioners are to be found in the Solution Development Team and should ideally be open at intellectual and experiential levels, be skilled in candid debate and capable of constructing realities based on their own perspective.
- *Knowledge Engineers*. These individuals belong to the middle management layer whose role in relation to knowledge creation is to facilitate knowledge along epistemological and ontological levels. This means promoting spiral learning (see Fig. 10.4) around all four modes and upwards from the individual, through the group and ultimately to the organisation. The skill set necessary to achieve this relies on project coordination and management capabilities, ability to determine hypotheses for new concepts, capacity to synthesize methodologies, communication including an ability to employ metaphors, engender trust and envision future courses of action (often based on past experiences).
- *Knowledge Officers*. Individuals belonging to the ranks of senior management must contribute to knowledge creation through the articulation of concepts and the knowledge vision thereby setting the standard for the justification of the value of knowledge (e.g., through the use of quantitative and qualitative criteria). Thus it becomes possible to pull together disparate themes into a coherent and moderately consistent view of the organisation with which activities can be guided and directed. This requires adoption of an equivocal stance that enables self-organised groups to autonomously find expression for the vision in more concrete forms. Clearly the capability to articulate, communicate and justify a knowledge vision along with fomenting creativity, gaining commitment from individuals and directing, managing, delegating and where appropriate facilitating activities are central to this role.

10.4 Self-Organisation

Self-organisation,[5] which is considered a critical success factor in agile projects [45, 115, 254], is characterised by teams that organise and manage their own work in an environment of participatory decision-making. This involves delegation of a liberal nature leaning towards as much autonomy as a team can handle. Measured on a

[5] For the purpose of discussion in this book, self-organisation is considered a synonym of empowerment. Self-organisation in agile teams has been researched in a surprisingly asymmetric manner with most research focusing on XP teams and comparatively little being done on other methodologies (e.g., Scrum, DSDM, DAD, SAFe®).

spectrum of direct control to complete democracy, however, there are several layers of delegation including command driven, persuasive efforts on the part of management without real influence of staff, consultative approaches where management seeks input, participatory measures where decision-making is delegated within limits and delegation involving minimal specification that affords a team considerable freedom. In order to thrive self-organisation requires that teams focus on a common purpose and language that endears a strong sense of self responsibility within a culture of mutual respect and trust and which enables constructive discussion of issues and challenges facing the team. Self-organisation implies neither complete democracy nor a lack of leadership, but rather captures a collective spirit of decision-making and resource management. There is a specifically agile connotation to self-organisation that identifies emergent behaviour as something which arises gradually over time perhaps as a logical consequence of the human-centric rules that teams employ [50]. This contrasts somewhat with the ecology inspired perception of self-organisation as a vehicle for order out of chaos.[6] Notwithstanding, evidence for spontaneity within teams does exist in the strong influence of implicit roles on group dynamics and flow (see Chap. 4). This suggestion, that complex interactions leads to emergent order is originally derived from the belief that self-organising teams are informal and temporary, arise spontaneously around issues, do not belong to formal organisational structures, possess members with strong sense of (common) purpose who determine their own affairs and whose primary roles relate to the task at hand. This perspective challenges and extends the rigid concept of project teams by introducing a more fluid and dynamic understanding of the organisation.

Self-organisation is tightly linked to the notion of a learning organisation wherein appropriate levels of redundancy (e.g., sharing of skills and knowledge) facilitate communication and collaboration. This requires adaptability, openness and a willingness to learn and change on the part of team members. Furthermore, self-organisation has been reported to be a feature of organisations that have rather flat hierarchies in which a broader scope of responsibility is shared by both managers and teams. In general, organisations can be categorised as having high hierarchy and low scope of responsibility (i.e., many levels with each manager overseeing a small number of direct reports) or low hierarchy and high scope of responsibility (i.e., few levels, but larger teams assigned to managers). Self-organisation tends towards the latter, but relieves managers to a certain extent of the burden of managing large teams. This is achieved by establishing common understanding of corporate vision from which teams as a collective can determine their own work schedules within the context of informal and decentralised structures that enable participative and democratic controls [17]. The following four principles, see Fig. 10.5, govern self-organisation in such organisations [113, 186]:

- *Minimum Critical Specification*. This refers to the specification of the critical factors required to direct a team and the placement of the least amount of restriction

[6] Such perceptions of self-organisation find their most potent expression in natural self-ordering systems and chaos theory.

Fig. 10.5 Self-Organisation Principles Published with kind permission of © Alan Moran 2015

on the team in order for it to function. In this context the balance of autonomy (described in detail in Chap. 11) places a crucial role.

- *Redundancy of Function*. This refers to the ability of individuals to engage in multiple different functions which requires the exchange of information and sharing of skills thereby establishing a link to the learning organisation [256]. It presumes that supporting the development of T-shaped skills (see Chap. 4) is desirable and that specialist individuals are less common.
- *Learning to Learn*. This refers to the capacity of the team as a collective to self-reflect on its practices and to adapt according to changes in its internal and external environments. This is most evident in practices such as daily stand-ups and retrospectives that act as a stimulus for group learning.
- *Requisite Variety*. This is concerned with the match between market segment volatility and the organisational unit assigned to address their needs. This concept promotes the notion that it the adaptability of self-organising teams that provides the best balance between the two. By corollary highly stable environments with little fluctuation do not necessarily need self-organising teams at all.

Self-organisation is understood as a knowledge creation agent in which learning occurs between groups and across functions within an organisation, requires low external and high internal autonomy (see Chap. 11 for a discussion of the three levels of autonomy), finds strength in diversity of perspectives, skills and behaviours. Interestingly whilst it is claimed that high individual autonomy interferes with self-organisation [180], self-organisation can in fact be found to operate well in both highly individualistic cultures as well as those that show greater deference towards authority. Examples of the wide applicability of self-organisational patterns can be found in individualistic cultures such as that found in New Zealand which scores

high on the Hofstede Individualism cultural dimension as well as India which whilst exhibiting moderate Individualism reflecting a mixture of collectivistic and individualistic traits also scores rather high on the Power Distance dimension suggesting a strong deference to authority [117]. These findings contradict the broad claims of the impact of individualism on self-organisation [111, 180].

As in any project, the distinction between customer and user must be made clear from the outset. The customer representative is defined as the individual who reflects the executive decision-making capacity in relation to the solution (i.e., they are responsible for setting scope and requirements, are to be held accountable for solution governance and must ultimately fund the undertaking). The users are those whose product interests the customer is representing and which must presumably be balanced with other wider concerns (e.g., strategic orientation, cost controls). Irrespective of the alignment of the customer and users, it is with the former rather than the latter that the agile team works.

Customer involvement is repeatedly cited as an issue in self-organised teams which suggests an ambiguous attitude towards commitment to product development arising from the tension between strategic concerns and the attending to of daily business [114]. Whilst agile makes very specific demands of the customer (e.g., on-site presence, committed, authoritative, knowledgeable, representative), it is to be expected that in practice varying degrees of collaboration are to be found ranging from on-site customer, remote access customers to team-based customer proxies. There are several reasons for this state of affairs including misconceptions concerning Agile on the part of the customer. Common misconceptions concerning Agile including the belief that changes are possible at any point during solution development (incl. within an iteration) resulting in poor customer discipline and involvement [114], lack of time commitment or ineffectual customer representatives, organisational distance,[7] the nature of the project terms of reference and imposition of traditional practices on the team. The imposition of traditional practices on an agile team is most common in situations where the customer is a large organisation contracting an agile solution team. In such situations the customer may be unwilling or unable to adapt its approach to the culture of its contracting partner. This is yet another example of how organisation can inhibit agility and innovation. Agile misconceptions are particularly troublesome because of the expectations that the customer may bring to the table and the behaviours that these encourage (e.g., the belief that being agile is a license to introduce arbitrary changes at any point in time or to dispense with planning activities entirely). Time commitment is significant and often underestimated since the customer in the agile environment is expected to commit resources to planning (e.g., one half-day approximately every two weeks), participate in regular reviews (e.g., one hour at the end of each iteration) as well as be on call (and preferably on site) for ad-hoc queries throughout. Furthermore, the customer representative must be sufficiently knowledgeable and in a position to speak with authority concerning

[7] Organisational distance refers primarily to the separation between the technical solution team and the operational business sphere of customers and users. This is particularly the case where the solution team is an offshore entity geographically located far from the customer.

the direction and scope of the solution. DSDM roles such as Business Ambassador supported by the Business Visionary, Business Analyst and Business Advisor (see the Appendix C "DSDM Roles") capture these requirements well and show the extent to which the business is embedded within the solution development effort. Accordingly, the assignment of junior personnel (lack of authority and knowledge), staff who do not adequately reflect the user base (lack of representation) or senior individuals who cannot commit sufficient time (lack of commitment) all threaten to inhibit flow and undermine the agile model. The result of this lack of effective engagement leads to lack of clarity identifying and prioritizing requirements and loss of productivity arising from the inevitable misalignment of solution and needs. Finally organisational influences (e.g., separation of staff by function, application of rigid standardised processes and terms of reference) all effect a level of external autonomy that impedes the development of self-organisation in teams. The following list describes the most common strategies for tackling customer involvement and engagement [114]:

- *Change Mindset*. This strategy addresses the need to challenge existing perspectives of the customer in a bid to persuade them to embrace a more agile approach (e.g., promoting agile notions of early delivery of business value). This belongs to customer education and is already widely used by agile coaches.
- *Build Confidence*. This involves enabling the customer to experience agile in a limited manner allowing them to build confidence before committing themselves fully to the approach. Contractually the experimentation with Agile can be reinforced with exit clauses should the desired outcomes not be reached. This strategy relies heavily on timeboxing which helps contain the risks involved and is considered a virtuous dynamic when see from the systems thinking perspective (see Chap. 1).
- *Readiness Prioritization*. This approach endorses the practice of demoting requirements when there is insufficient information. For agile methodologies that use forced ranking systems[8] (e.g., Scrum, DAD, SAFe®) this amounts to moving items further down the list and preventing their transition to an iteration until they are sufficiently clear. The MoScoW prioritization technique of the DSDM Agile Project Framework, already offers an alternate approach to such matters.
- *Project Approach Questionnaire (PAQ)*. This requires that the customer assess their commitment and understanding of the approach early on in the project and is already a codified and well established practice in the DSDM Agile Project Framework. For example, some questions in the DSDM PAQ directly query the ownership of the project, understanding of MoSCoW prioritization and the importance of customer involvement and its consequences (e.g., access to Business Ambassadors, Business Advisors and Business Analysts).
- *Owner Partitioning*. This strategy is essentially a refinement of the Scrum Product Owner role wherein requirements are partitioned and assigned to one of several owners. This effectively spreads the burden of customer representation amongst

[8] Forced ranking systems employ a product backlog wherein items towards the top of the list have higher priority and are specified in greater detail to those found lower down on the list.

individuals who can better support each other and ensures a more even input into planning. This approach, however, does require some degree of oversight and coordination to ensure a consistent and cohesive level of product planning.

- *Customer Proxy*. A rather pragmatic, though common approach to tackling customer involvement is to nominate an individual within the solution team to assume responsibility for coordination with the customer. Such an individual requires merely an understanding of Agile from the customer perspective (e.g., formulation of requirements as user stories). The risk borne by this approach is that it may legitimize the lack of direct involvement on the part of the customer (e.g., less involvement of the Business Ambassador, Advisor or Analyst).
- *Progress Reviews*. The presence of the customer at the end of each Timebox in order to review solution progress and assess the readiness for release of increments is a vital element of involvement and one that is often welcomed by customers. This provides an interim point at which to detect any departures from expectation and to limit the impact that these may have (i.e., to offer early corrective advice) and has been found to assuage even the most sceptical of customers. Reviews are a common feature of almost every agile methodology.
- *Virtual Collaboration*. This strategy alleviates some of the issues related to customer collaboration particularly when a lack of co-location adds to time costs of attending agile events. For example, few customers are likely to want to travel to attend a short stand-up meeting, but would find it acceptable to make a video link appearance. This strategy also extends to a variety of other transparency enhancing practices (e.g., electronic information radiators and burndown charts or the use of chat to solicit quick feedback from customer representatives).

Table 10.2 describes how these strategies relate to the causes of customer apathy and illustrates how several approaches may be employed in concert to improve customer involvement and engagement.

Table 10.2 Applications of Strategies to Tackle Customer Engagement Issues

	Agile Misconceptions	Organisational Distance	Time Commitment	Institutional Customers	Ineffective Representation
Change Mindset	X			X	
Build Confidence					
Readiness Prioritization			X		
Project Approach Questionnaire			X		
Owner Partitioning	X		X		
Customer Proxy		X			X
Progress Reviews	X	X	X		X
Virtual Collaboration		X			

Adapted from [114]

Turning the attention to managers, these need to be sensitive to the organisational needs of Agile and understand the influence of self-organisation on learning, innovation and productivity [112, 193]. For example, the impact of hierarchy is both to create distance between decision makers and those with the operational knowledge needed to inform them and to overburden the communication process. The terms of reference for projects are largely influenced by the decisions of managers who may choose to act in a discretionary manner (e.g., dispensation concerning use of standardised processes or tools, intervention during contractual negotiations to ensure that Agile is accommodated). For example, fixed-pricing, which is framed by traditional notions of project management in relation to control attributes such as time, scope and quality is less appropriate for agile projects that anticipate the uncertainty which undermines estimation. This means that either the pricing model needs to be challenged (e.g., accepting that scope is a variable) or amended (e.g., adding a risk buffer for unanticipated changes in the original specification). The premise on which such intervention is based is warranted on grounds of risk management (e.g., responsive to changing environments) and project governance (e.g., prioritizing the delivery of business value) and is considered a key success factor of agile projects by the DSDM Agile Project Frameworkand wider research findings. For example, the view that a conducive business environment, as described in Chap. 4, is important is reflected in research findings [112].

Agile projects frequently require specific infrastructure in order to operate effectively (e.g., information radiator, continuous integration and automated build environments, deployment pipelines) as well as support personnel (e.g., coaches, facilitators). Therefore the necessary resourcing needs to be in place in order to ensure that teams have the support they require and have someone to turn to in times of need. Whilst teams often engage themselves in the erection of simple infrastructures., they frequently feel rather overburdened with the scale and breadth of initiation (e.g., training of the customer in Agile and expectations concerning their role therein). Instead, this should really be thought of as a function of an agile project management office with the capabilities to instantiate working environments and on-board stakeholders. Additional resources may be required to tackle specific project issues as needs dictate (e.g., distributing mobile devices to business stakeholders who would otherwise be unwilling or unable to time commit to project events such as daily stand-ups).

Another domain in which agile managers have a role to play is in the resourcing of agile project teams not only from the point of view of their constitution and mix, but also their stability which is regarded as yet another critical success factor [112] of agile projects. These elements combine to ensure sufficient internal autonomy and cohesion for collaboration to thrive. Specifically an agile manager should ensure that teams are sufficiently diverse and robust (e.g., to protect against group think and fragmentation tendencies) and also guard against lending out team members to other projects thereby weakening stability and group identity. It follows that a performant team which has completed a project should ideally be kept together as a unit for the next project. Indeed retaining team integrity for high performance teams makes a lot of business sense since customers generally expect a sustained level of performance

from one project to the next and are likely to want to keep the same team together. Self-identity that is forged within such a group can nurture high levels of commitment and motivation so often found in self-organised teams.

These findings suggest that the agile managers must either ensure that informal structures are established or that they can at least co-exist within line management organisational units in such a manner that issues may be raised without fear of reprisal or criticism and in an atmosphere of openness and trust. The management implication of this includes adapting the hiring process to find individuals who fit not only in a technical, but also in a social sense and engaging the team to find appropriate new members. Evidence-based management interview approaches are particularly well suited in this regard since they offer a genuine opportunity to assess candidates in action (e.g., scenario-based interviewing).

If self-organisation, often introduced with the objectives of improving effectiveness and quality of working life, is instrumental to enhancing organisational flexibility then it requires a fine balance between management direction and team autonomy [159]. Though this empowerment can in principle be achieved through individual job enlargement and cross-training, a rethinking of how control is exercised, how teams are developed and how their boundaries with the external environment are managed is called for. Barriers, however, still remain [19]:

- *Opportunities*. The level of management support that management is willing to provide thereby relinquishing direct control in favour of coaching and servant leadership. This can present a considerable challenge to managers who must forge new identities or fear loss of position or status. Some may engage in subtle spirals of assuming increasing responsibility on behalf of the team leading to ever more passive team attitudes that erode the effectiveness self-organisation and degrade team performance.
- *Attitudes*. The psychological needs of employees which may be influenced by a number of factors. For example, the assumption of new responsibility (e.g., self-organisation paradigm) may trigger an expectation for higher rewards or specialist skills may foster a sense of uniqueness and importance linked to self-identity (e.g., status). The relationship between responsibility and reward features strongly in the realm of equity theory.
- *Abilities*. The skills and learning abilities of employees both in terms of technical skills and social abilities which may limit the extent to which self-organisation may flourish. For example, limitations in this respect influence the extent to which tasks can be rotated or knowledge shared.
- *Requisite*. The extent to which self-organisation is actually called for which reflects the nature of change in the external environment (i.e., requisite variety). Thus in situations where routines prevail then efforts towards standardisation may provide more efficient than self-organisation.

Thus managers must take account of several elements, including their own behaviours, that impede self-organisation. Of these the requisite is the most dominant being capable of driving or limiting the extent of self-organisation. Thereafter, depending on the nature of the organisation, opportunities can become the limiting factor leaving only attitudes and abilities as determinants.

10.5 Scaling Agile

Scaling agile (e.g., a programme comprising of several related and perhaps concurrent projects) has consequences for how teams are organised since so much of Agile relies on effective communication and collaboration which in turn implies colocation of small teams that employ face-to-face channels. To an extent technology can support larger groupings (e.g., teams of teams) and becomes essential when these are geodispersed. This results, however, in a higher reliance on tools (e.g., electronic information radiators and web-based planning tools) that may either mildly inhibit activities (e.g., the need to locate and authenticate prior to use) or legitimise overheads (e.g., reporting requirements) that do not add any essential value. Furthermore, scaled agile practices introduce new roles, strategies, practices and artefacts necessitated by the wider scope and complexity of the undertaking and thus it would seem that scaling is sometimes inescapable therefore prompting the question of whether or not agile organisational structures are capable of scaling.

The most commonly cited topology involves using project levels teams and creating impromptu boards comprising of representatives of each of the teams to discuss and coordinate intra-team activity. In the Scrum methodology, this topology is sometimes referred to as a "Scrum of Scrums" and the representative may be any individual taken from each Scrum team. Individual teams may be organised around either subsystems or requirements channels [75], but thereafter kept as stable as possible throughout the change programme. Augmenting these teams there should also be support teams to tackle issues at the programme level (e.g., integration and deployment, high-level architecture, methodological expertise). Rather more challenging is the cohabitation of programme and project structures with more fixed organisational units (e.g., departments) whose culture may be more silo aligned. Indeed, this introduces a clash of change oriented organisation versus traditional hierarchical structures whose resolution depends in large part on the overall agility of the organisation. In the interests of the programme, it would suffice for programme level autonomy to be protected and translated downward into project level autonomy within the existing support structure. This requires considerable management support, coordination, organisation and leadership.

A lot of agile practices remain unaltered in the sense that they are conducted as before within small groups of individuals. Thus pair programming still only requires two individuals and stand-up meetings or retrospectives reflect on the working dynamic of project teams. Scaling of practice occurs only when it makes sense to do so. For example, facilitated workshops (e.g., during scoping and

requirements gathering) may involve very large groups where the facilitator is supported by several co-facilitators or continuous integration and deployment may assume the dimensions of system integration and automated end-to-end testing. The extra investment in organising and enabling such practices represents the cost of scaling and of course must be justified by the circumstances (e.g., projects that have dependencies or interfaces). Few methodologies give specific and detailed advice on how scalability is achieved, though the Scaled Agile Framework® (SAFe®) is an exception in this regard. SAFe® envisages teams gathered under the auspices of a programme that delivers increments using the shared services of deployment and delivery (e.g., DevOps) that can be embedded into agile teams to ensure continual production readiness thereby reducing lead times, integration issues and delays. Other support is provided by user experience (UX) tracking, at the programme level, ahead of the project teams in order to centralise design (but not its implementation) or the provision of an agile scalability expert (agile scalability experts are referred to in SAFe® as a Release Train Engineers) at the programme level. All this takes place within the context of continual reassessment and reprioritisation of the programme backlog and application of lean economics to determine job scheduling practices in order to maintain overall flow. By contrast DSDM, which has a broader scope than the architectural and product focus of SAFe® encourages the incremental delivery of software capabilities using programme and project level support structures (e.g., Portfolio Management and Support Offices) whilst leaving open the specifics of team organisation and topologies beyond citing a few roles that need to be occupied.

One often overlooked aspect of scaling is the learning that must take place above and between project teams. Given that feedback and learning are so important within project teams it is legitimate to ask how this is reflected higher up in the scaled organisation (e.g., at the programme level). Since it is infeasible for each team to integrate with every other team, it falls to management understood as knowledge engineers and the agile support infrastructure to become the custodians of wisdom and best practices. For example, a coach servicing the needs of several teams gains insights into what appears to be working well in some teams and may act as a catalyst to encourage other teams to experiment along similar lines. On the other hand, management may be in a position to establish and promote a programme level knowledge base with which to ensure that combination activities (see Fig. 10.3) take place. Management may also consider programme configuration management (see Chap. 9) to be a learning device with which projects can gather feedback and compare the results of their activities with other teams (with the proviso that this be a device that remains under the control of those affected by it).

10.6 Management Implications

Managers should not fall into the trap of believing that mere practice of agile techniques (e.g., daily stand-ups) constitute an agile transformation. Based on a deeper cultural analysis of Agile, an effective strategy must address all three levels of corporate culture. This means that at the deepest level there must be a presumption of competence and ability to assume personal responsibility amongst staff who must be able to reasonably assume that adaptation and collaboration are a given within the organisation which actively supports learning and continuous improvement. This must be reflected in a palpable sense of innovation in which moderate levels of risk taking are acceptable and may be engaged in without fear of reprisal (e.g., blame culture). In other words: better to have risked and lost than never to have risked at all should be the credo. These underlying assumptions must incorporate the self-evident belief that Agile is in the interests of all stakeholders and aligned with corporate strategy. Atop of these basic assumptions are more visible manifestations of Agile in the form of self-organising teams, embedded customer representation and underlying values of trust, integrity, honesty and openness. This is where middle management can play an exemplary role by demonstrating appropriate behaviours and reprimanding unacceptable ones. This involves active fostering of a value system that must be interpreted in terms of daily activities and decisions through which agile norms are established. Support networks and infrastructures become an essential part of building up and sustaining agile capabilities (e.g., agile coaches, system administrative support for tools). The final level is perhaps the easiest since it involves the enacting of agile practices within the belief and value system and basic underlying assumptions already described above. Ironically, this is where most agile consultants begin only discovering later the folly of trying to establish such practices on infertile ground that believes, values and assumes a world view that is at odds with the agile philosophy. This most visible level of practices and artefacts is precisely the easiest to implement because its visibility affords it direct and immediate feedback on the adoption of Agile.

Organisational learning assumes strategic significance when it is seen as that capability to create, assimilate and employ new knowledge which requires that the senior management of an organisation assume a stance on what the knowledge vision should be and underpin it with the necessary management structures to support it. This includes involvement of employees which in turn requires ensuring they possess sufficient empowerment and autonomy in order to be act to act with initiative and personal responsibility. Such a commitment requires establishing self-organisation as the primary organisational unit and ensuring that teams are diverse in terms of experience, skills and outlook. Furthermore the organisation as a whole needs to adopt a sense-and-adapt attitude towards its external environment. Indeed, it is precisely in turbulent environments that agile teams excel since they are able to act appropriately providing that they are in fact capable of sensing their environment accurately. This necessitates a form of second loop learning wherein existing approaches are permitted to be questioned by revisiting the premises on which they are based through a

process of equivocality and reflection. In order to remain open to new experiences, a degree of redundancy in relation to knowledge is required (e.g., sharing of experiences and skills, use of information radiators and transparent reporting). Finally organisational structures must mirror the information fluidity needs suggesting that flat and flexible structures interlinked with informational channels appropriate to the nature of the external environment (e.g., news feeds on technology trends or moves by competitors).

Systems thinking appears to suggest that the integrated and generalist character of agile teams is a net benefit in terms of productivity, quality and delivery of features. Specific practices (e.g., refactoring, test-driven development and pair programming) when used in combination act to reinforce each other resulting in considerably fewer defects and an overall reduction of technical debt. Given the empirical evidence in support of these claims it becomes hard to justify the existence of separate functional units (e.g., testing) that incur higher coordination and communication costs and impair the timeliness of feedback which in part is key to building trust, team performance and business satisfaction. This suggests that management ought to consider carefully the impact of organisation on the dynamics of solution development.

What complicates organisational change, especially cultural aspects, is perhaps less the change itself as the manner in which it is performed. Change programmes need to be honest and holistic and conducted with a sense of purpose and integrity that involves the views of those affected. This calls for leadership from the top of the organisation as well as support (e.g., coaching) at all levels. The classical ingredients of change management [146] (e.g., a sense of urgency, building coalitions, creating visions and enlisting others, removal of barriers, establishing quick-wins, sustaining and institutionalising change) still apply to agile initiatives. However, agile environments have the capacity to sustain continual change owing to their inclusive and retrospective approach. Transitioning from traditional vertical skill sets (e.g., software programmer, database administrator, requirements engineer) can pose a considerable challenge particularly for those with long experience of working in such environments.[9] This passage towards a broader range of skills and activities is fraught with resistance arising from conflicts of self-identity, feelings of personal criticism and mistaken beliefs that redundancy is the antithesis of efficiency. It also imposes new demands on individuals (e.g., social competencies) who may up to that point have relied largely on their functional and technocratic skills within the team. Indeed, the selection of individuals based not only on their technical, but also their social abilities does feature in mature agile environments where the entire team may be involved in the hiring process [113]. Accustomed to command-and-control structures many managers themselves experience significant issues with the relinquishing of authority to the team. This requires trust on part of the manager who must instead focus on establishing a culture that supports individuals, their interactions, communication and collaboration and finding new responsibilities in the promotion of spiral learning. In particular, the classic perception of a project manager as controller and planner must

[9] Experience appears to suggest that transition issues are more common amongst senior rather than junior staff.

be replaced with one of facilitator and collaborator [193]. The agile manager must therefore be attuned to the dynamics of the team (incl. customer involvement) and be prepared to engage accordingly where appropriate. Observance of the external environment is also central to the manager's remit since the extent to which self-organising exists, enhances the adaptability of an organisation. Failure to address this balance will result in an enterprise whose organisation itself is becoming its limiting factor.

Chapter 11
People Factors

Abstract The psychological make-up of agile teams has been the subject of relatively little research in spite of the emphasis on humanism and emotional intelligence in agile management. Given the clear socio-technical character of the agile process, it is important to establish a broad understanding of its implications for teams and their performance. It would seem that a variety of people factors (i.e., motivation, commitment, emotional intelligence and personality) are intimately linked. Motivation and commitment thrive in the self-organisation paradigm of agile environments and related practices, benefiting from the autonomy and purpose found therein. In this context emotional intelligence plays an important role not least for managers who must develop it as part of their competencies if they are to be able to successfully manage agile teams. In terms of personality the message that diversity is a strength is confirmed in multiple empirical studies. There are two complementary approaches to this topic that furnish some insight into the relevance of personality to Agile. One is based on descriptive aspects of personality whilst the other an attempt to explain potential mechanisms behind it. Finally, it is possible to gleam from the sparse, but tantalising research some general principles that find application in daily management.

11.1 Introduction

Accustomed to perceiving ourselves as purely rational thinkers, it is easy to disregard the influence of feelings and emotion in our actions. Cognition is concerned with gathering and assessing data in an environment within an existing belief and value system from which assumptions stem. Feelings, on the other hand, are affected by social context and mood congruence.[1] However, it is the interplay of cognition and feeling that drive behaviours and hidden behind this visible manifestation of identity are people factors i.e., a combination of motivation, commitment, emotional intelligence and personality. This is of importance not only to oneself and the ability to develop self-awareness, but also to the understanding of others (e.g., tackling

[1] Mood congruence refers to the consistency in which emotions that are felt are actually being displayed in a visible manner.

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1_11

cognitive bias[2] and transference[3]). These competencies are of particular importance in agile teams where social and interpersonal skills play at least as important a role as technical ones.

Self-organisation was identified in Chap. 10 as the central organisational paradigm for agile teams. Self-organisation relies on team flexibility, autonomy and diversity, which in turn require adaptability and emotional intelligence on the part of team members. This link to self-organisation is crucial for an understanding of people aspects of Agile since this is the arena in which these factors find expression and creative tension. Flexibility, autonomy and diversity have been found to contribute positively to overall performance as measured in terms of product features, time to market and market success as discussed later in this chapter. Accordingly there emerges a picture wherein personality becomes a key contributor to the success of agile (e.g., ability to resolve conflicts and adapt to situations within a self-organising context) within organisations that are willing and capable of adapting themselves appropriately (e.g., innovative and open cultures, flat hierarchies and fluid organisational structures). Furthermore, motivation and commitment is found to flourish in feedback-driven environments where autonomy and purpose lead to stronger identification with a project or organisation. Note that autonomy ought not to be equated with independence or individualism. As shall be seen later autonomy is manifest at many levels and concerns the locus of causality of activity (i.e., whether it is externally driven or internally inspired). Furthermore, autonomy is conditioned by ambient demands, rules and regulations and the sociocultural context within which activity is performed (e.g., what is expected of individuals and the extent of their personal compliance towards the demands and wishes of others).

Understood as a socio-technical process, Agile poses new demands on the members of solution development teams whose traditional sources of identity (e.g., analysis and design skills, technical prowess) must be augmented with new social and interpersonal skills. These latter elements are rooted in personality and reflect enduring traits that distinguish individuals and their behaviours. There is already much evidence that the personality profile of a team contributes towards performance [99, 144]. Despite the research, however, there is little acknowledgement of this fact within agile communities with some notable exceptions [49–51]. For example, social and interpersonal skills do not feature amongst the critical success factors of the DSDM Agile Project Framework, which considers business knowledge and technical expertise together with good communication skills as sufficient. Arguably, less important than individual personality traits, however, is the mixture of personalities within a group. In particular, certain combinations of traits are found to be desirable whereas the presence of other traits can be detrimental to overall team performance. Since it appears that heterogeneous teams are generally better at solving

[2] Cognitive bias refers to irrational inference and judgement based on socially constructed realities that omit to take into account relevant objective inputs.

[3] Transference refers to the transfer of a pattern of behaviour in relation to one individual over to another in an inappropriate manner. It is most evident is the passage of inappropriate childhood relations into the adult world.

unstructured problems (e.g., explorative searching for solutions that best fit emergent requirements) this appears to indicate that differences in personality compensate for individual limitations within the team [277]. The agile manager must therefore be aware of the strength found not in diversity *per se*, but in specific forms of diversity.

That psychological needs are linked to tasks is evident in the fact the individuals who express need for growth thrive when working on complex jobs. Particularly in knowledge worker environments, it is true that learning in which individuals experience personal responsibility in the course of execution of a task about which they care results in positive effects and that are reinforcing and which incentivise individuals further. Thus some dimensions of work (e.g., skill variety, task identity and significance) contribute towards meaningfulness whereas others (e.g., autonomy and feedback) fulfill other psychological states such as responsibility and knowledge of the outcome of activities. This in turn promote motivation, quality, performance, satisfaction and commitment [98]. This would suggest that knowledge workers generally require variety in their work over which they have control and whose impact and outcome is deemed significant and made known to them. This is perhaps why agile heterogeneous teams are often so motivational since they offer direct lines of communication and feedback to a solution development team and afford opportunities to identify with outcomes of significance.

11.2 Team Psychological Safety

Of central importance to agile teams are those factors that enable team learning. Traditional thinking dictates that a focused task, appropriate construction of teams and access to the necessary resources are all that is required in order to ensure performance. However, understood from the perspective of organisational learning (see Chap. 10) cognitive and interpersonal skills emerge as important factors particularly in innovation environments where change and uncertainty prevail. Organisational work related to learning is characterised by feedback, information sharing and experimentation, which helps establish a common understanding and uncover issues. This necessitates trust since the ability to admit to an error makes an individual vulnerable in respect of self-esteem or the appearance of incompetence thereby introducing intellectual rigidity that may erode the value generating potential of the group. Indeed, openness is vital if adaptive behaviours that adjust the course of a project (once false assumptions or incorrect action have been uncovered) are to be supported. Such attitudes are deeply linked to organisational culture, tolerance of risk and how learning is practiced and perceived within the organisation. The term team psychological safety was coined to reflect the shared belief that it is safe within a team to take interpersonal risks, a notion that is unrelated to the cohesiveness of the team.[4] Thus team psychological safety refers to the sense of confidence that a

[4] The phenomenon of group think is an example of how highly cohesive teams may also exhibit high levels of interpersonal risk.

team will not embarass, reject or punish an individual for raising questions. It is not sufficient for individuals to be amenable to taking such risks, since the team must demonstrate through shared experience their trust and respect in situations where errors occur. This is confirmed by other findings that illustrate how attuned individuals are to relational matters (e.g., how they are regarded by their social group) rather than other forms of sanction (e.g., line manager disapproval). Team psychological safety alone, however, is not sufficient since the individual must believe that by admitting to an error the team is capable of using this information to good effect (e.g., plotting a new course of action in a sufficiently responsive manner).

Necessary precursors for team psychological safety include a clear vision in relation to the what is to be achieved in a project, the support for the team (e.g., access to resources and information) and the role of team leader as coach in directing activities (and whose exercise of authority will set the tone for the team). This means the structural aspects of a team impinge on the beliefs held by the team which in turn influence team learning behaviours (e.g., the ability to solicit feedback from others, discuss errors or consult with customers) which ought to be manifest in team performance as measured by customer satisfaction with outcomes. This is validated by research that indicates that learning behaviour is a significant predictor of team performance and that team psychological safety, understood as an inherent belief system in the good natured intentions of others, not only exists but is strongly associated with team learning behaviours. It would seem that the dominant consideration is how others react to situations in which an individual undertakes a course of action whose outcome is uncertain (and may result in failure). This points to socially conditioned attitudes towards risk and its management (see Chap. 8) as well as the relative unimportance of team efficacy (i.e., how well the team responds in such situations). Moreover team leadership and team support whilst contributing positively to the conditions necessary for team safety and performance do not entirely shape the emergent beliefs within the team. Rather it is the mediating effects of team psychological safety that bring about the team learning that delivers performance.

11.3 Motivation

Motivation in its simplest form has been found to thrive when mastery of inherent abilities and skills can be directed towards a meaningful purpose in an autonomous manner. These key motivational ingredients of mastery, purpose and autonomy when supplemented with the emotional intelligence factors of personal growth, positive relations with others and self-acceptance constitute a rather complete model of wellness both in personal and occupational life [204, 221]) though other researchers suggest that a more complete list is required to account for wellness [222]. Mastery is an individual capacity that grows and spreads in a culture of organisational learning (see Chap. 10 for further details) whereas purpose and autonomy are found to develop when there is an actionable identification with the goals of a project or organisation.

Both of these latter aspects are the topics of this section and are repeatedly referred to elsewhere throughout this book.

Motivation describes the forces that account for level, direction and persistence of effort expended in the pursuit of an endeavour or goal. Although there are several theories concerning motivation, empirical evidence for many of them appears to be rather mixed suggesting that motivation is subtle and complex[5] [7, 105, 163, 167, 264]. There are nonetheless some general remarks of relevance that can be made which describe both virtuous and vicious performance cycles, derived from a combination of motivation, individual capabilities and the support received from the organisation which in turn shapes the future development of motivation. As illustrated in Fig. 11.1 (wherein dotted and dashed lines indicate routes to positive and negative impact on motivation respectively) an individual endowed with personal needs, values and expectations embarks on an undertaking that is subject to constraints (e.g., limitations on resources and funds, skills, time). During the course of this undertaking the aim is to achieve specific individual and organisational goals on whose outcomes (i.e., success or failure), there arises either positive or negative reinforcement[6] which in turn affects the level of motivation and willingness to expend further effort.

From this model several important factors for motivation in agile environments become apparent including the nature of the constraints that an organisation imposes on a team (e.g., level of autonomy, decision-making models, impact of organisational structures on access to resources) and the alignment of individual and organisational

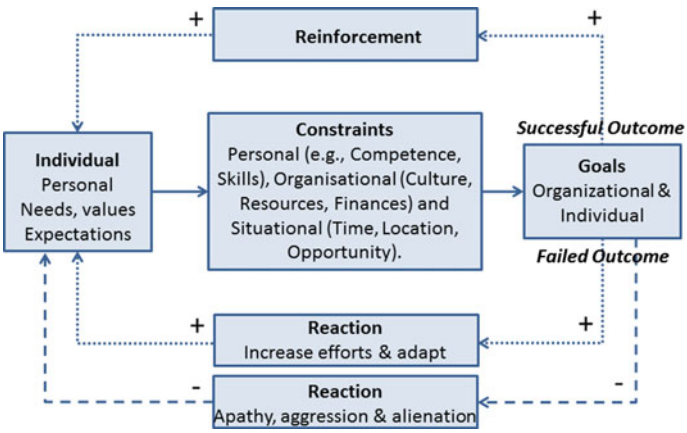

Fig. 11.1 Motivation Cycle Model. Adapted from [85]. Published with kind permission of © Alan Moran 2015

[5] The common theories of motivation can be classified into content theories that address needs (e.g., Maslow, Alderfer, McClelland, Herzberg) and process theories, which focus on levels of satisfaction (e.g., equity, expectancy). Empirical evidence for motivational theories is, however, rather mixed.

[6] Positive reinforcement need not be exclusively tied to success, since even failure can inspire an individual to redouble their efforts and try again.

goals (e.g., identification with a project and its outcomes). Positive motivation from successful outcomes is born of acknowledgement and recognition, which in agile teams is usually attributed to the group and its ability to self-organise towards a common goal (e.g., success in establishing common understanding and resolving conflicts), thereby enhancing its sense of purpose rather than being dispensed to individual heroes. Such acknowledgement need not necessarily be handed down by management but may also arise through recognition from within the group thereby strengthening social bonds. Reactions to failure, on the other hand, are influenced by both personality (e.g., neurotic responses versus deliberate conscientiousness) and organisation culture (e.g., blame culture versus risk-taking innovation), but also have an agile dimension (e.g., retrospective sessions).

Motivation is often classified as either intrinsic or extrinsic [220]. The former refers to the engaging in activities on the basis of their inherent interest and is not subject to external lures or pressures whereas the latter concerns the taking part in something because it leads to a separate outcome (e.g., tangible reward, avoidance of sanction, utility benefit). Motivation as the impetus to act is related to the task at hand and much of the research findings concerning motivation are predicated on an interest in that task. Therefore any discussion of mastery, purpose and autonomy becomes largely irrelevant if there is not an intrinsic interest in the task (e.g., challenge, novelty). It turns out that extrinsic motivation is considerably more stratified than intrinsic motivation and this variance is linked to the extent to which it is autonomous. For example, there is a difference between creating a design document out of fear (e.g., managerial threat of sanction) and writing it because it is the right thing to do (e.g., personal volition). Both are forms of extrinsic motivation since they relate to separate outcomes (e.g., avoiding a sanction or the utility benefit of having a design documented) but each reflects either a low or high level of autonomy. Thus what autonomy provides is the space and means to transform activities and values into their own. As indicated in Fig. 11.2 motivation spans a self-determination continuum comprising of the following elements [221]:

- *Amotivation*. This refers to the lack of value perceived in an activity which may arise from deficient mastery or efficacy (i.e., the belief that the outcome can be achieved). Autonomy will not help here since there lacks the capacity to exploit it.
- *External Regulation*. This captures situations in which external forces place demands (e.g., making threats or offering incentives). Thus behaviour is controlled and a sense of alienation can arise in this context.
- *Introjected Regulation*. There remains an element of control in which individuals are motivated to act out of a sense of pride or a desire to avoid failure. This is typically linked with feelings of self-worth and esteem.
- *Identification*. Though a form of regulation, the activity has become accepted and personal importance is attached to it. Here there is an increased sense of autonomy that is derived from ownership of the task.
- *Integration*. This occurs when there is high levels of volition wherein the task has been evaluated as consistent with one's own beliefs and values giving rise to goal

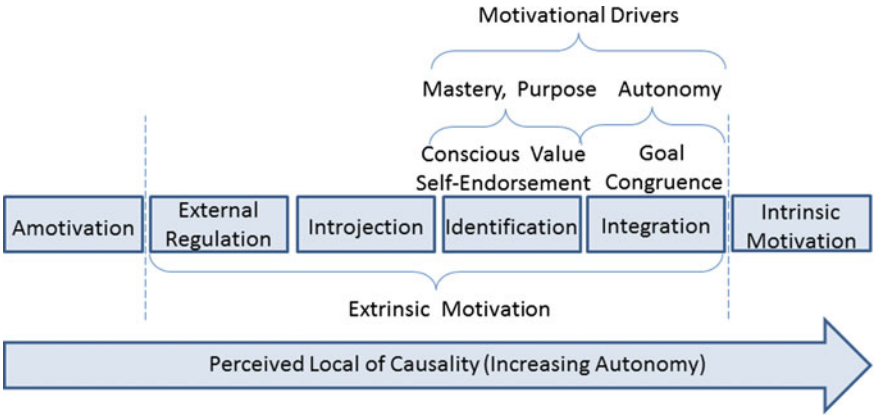

Fig. 11.2 Self-Determination Continuum. Adapted from [220, 221]. Published with kind permission of © Alan Moran 2015

congruence. The task may still be considered extrinsically motivated since it is not undertaken solely to satisfy a personal sense of enjoyment.

- *Intrinsic Motivation*. This describes the personal inclination to assimilate and master a task largely on the basis of its inherent attraction and the sense of joy derived from it without reference to outcomes that are separate from the execution of the task.

This continuum appears to indicate why repeated studies have shown that extrinsic rewards (a form of external regulation) erodes motivation since they reflect an external perceived local of causality and thus contribute towards a sense of diminished sense of autonomy. Indeed almost any attempt to externally influence affairs (e.g., threats or rewards contingent on performance) have the effect of reducing intrinsic motivation. For example, making reward contingent on performance as assessed by an external agent (e.g., manager or stock price of the company) immediately creates a disconnect with individual autonomy that is damaging for motivation. Equally accommodation of feelings and admitting self-determination have been found to enhance intrinsic motivation [221]. Thus whilst mastery and autonomy in combination facilitate improved motivation, purpose encourages security and relatedness provided that this occurs in a supportive and caring environment. Key to this development is identification and integration since this is where action entails personal endorsement and the transition towards volition occurs. In fact, prior to this point there exists less effort towards achievement and a tendency, when things go wrong, to blame "the system" owing to a lack of personal responsibility or a tendency towards self-serving bias.

If facilitating extrinsic motivation is important, then the question arises as to where this journey should being. The main reason why individuals initially engage in an action is because they it is perceived as valued by others and in particular by those held in high esteem by the individual concerned. Thus the trigger towards motivational

behaviour may be inspired by leadership whose integrity lies in the consistency of expressed value and action and the meaning that can be attributed to a specific course of action.

Since the motivation of a team is a reflection of their efforts towards what is perceived to be an important outcome for which some form of reward can be expected, motivational gurus often use the expectancy model which suggests that individuals have a choice on how they wish to behave and that they are motivated to opt for specific behaviours that either maximise desired expected outcomes or minimise expected undesirable outcomes. Such assessments rely both on their perceptions of the situation and on their values in terms of the following elements that are interpreted here in the agile context:

- *Expectancy*. The belief that increased effort will result in increased performance (subject to the availability of resources, skills and support). This judgement is based on a personal assessment of ability, the level of difficulty of the goal and the extent to which the outcome can be influenced by the individual. Though the classical literature describes expectancy from an individual perspective it is plausible that an individual's expectancy is buoyed up in the context of a support collective environment. Interestingly, as will be seen later, agile teams do tend to thrive in the face of challenging projects that require creative and innovative solutions.
- *Instrumentality*. The belief that performance will result in a reward that is meaningful to the individual. Rewards may be extrinsic (e.g., money) or intrinsic (e.g., satisfaction). Of equal importance here in the agile context is how reward relates to the group (e.g., sense of collective achievement). This is enhanced within an environment of procedural justice,[7] trust and transparency all of which need to be present and sustained in agile environments. Accordingly, instrumentality may be improved by group recognition and could be undermined by singling out individual heroes for reward.
- *Valence*. The importance (either negative or positive) placed by the individual on the expected outcome. In agile projects this may reflect the level of identification that individuals share with the project and its outcomes. Later in this chapter it is noted that it is the positive effect that the agile approach and attitude has on team members which contributes significantly to commitment and thus to valence, insofar as this translates into identification with the outcomes or attribution of importance to them.

Motivational force is defined as the (numerical) product of expectancy, instrumentality and valence and so a specific motivational level may be arrived via different routes (e.g., high expectancy, but medium instrumentality and valance or high expectancy and instrumentality, but low valence). Irrespective of the magnitude of the elements, only valence has the capacity to determine the orientation of motivation (i.e., negative or positive) which is then of course amplified by the product of

[7] Procedural justice refers to the perceived fairness and transparency of reward allocation (i.e., the rules of the game).

the other two. Together with the motivational cycle (see Fig. 11.1) these two models capture both static and dynamic aspects of motivation and provide some clues about the forces that are at work in agile teams.

The relevancy of expectancy in Agile is deeply woven into its culture and practices. Amongst other things, it extends the notion of projects being built around motivated individual[8] to the definition of appropriate incentives (e.g., flexible approaches customised towards at the individual and group levels), controls based on informal mechanisms reinforced by formal rules or principles and quality of working life (e.g., sustainable pace) framed as a motivational factor [22]. For example, adaptive and inclusive planning mitigates the sense of powerlessness that will be alluded to later in the discussion of commitment that would otherwise have eroded expectancy. Furthermore, the culture of openness and transparency found in agile communities helps build instrumentality (e.g., definition of done statements) and fosters trust between team members. In fact, trust can be found to explain variations in effort in problem solving owing to the comfort that individual find when working on group tasks in high trust environments [68, 158]. Indeed, the elements of the expectancy model become particularly clear in this context since working together creates a heightened sense of expectancy, increased instrumentality as a result of mutual recognition and transparency and enhanced positive valence that finds expression in reward of meaningful outcomes. In order that rewards (understood here in terms of recognition of work, satisfaction earned through delivery of solutions) be recognised, however, a sense of collective ownership needs to be established since within the agile context rewards are issued to the group rather than to an individual.[9] Accordingly, it is worth considering the incorporation of socialization events that celebrate the achievements of the team as a collective (e.g., release party on successful delivery of a solution increment).

Maintaining a sense of fairness surrounding reward structures within a group whilst continuing to pursuing optimal outcomes (i.e., the difference between rewards and the costs incurred acquiring them) is as important in agile environments as non-agile ones. In agile teams collective reward systems require a means of apportioning reward to group members and encouraging behaviours that are beneficial to the group as a whole. This implies that individuals will become motivated to rectify perceived imbalances in the ratios of outcomes to inputs and that violations (e.g., inequitable behaviours) will not be sanctioned. This requires conflict resolution abilities and emotional intelligence and a failure to apply these appropriately will come at the cost of instrumentality and ultimately motivation. Within the agile context intrinsic rewards (e.g., recognition, reputation, praise) may be allocated to the team as a collective whilst undesirable behaviours (e.g., social loafing) may be subject to critical social responses at any one of a number of feedback events (e.g., daily stand-up meetings). Whilst managers may feel that they must be the ultimate arbitrators in

[8] The suggestion that agile projects *per se* are made up of motivated individuals appears to be taken for granted in the agile literature pointing perhaps to a degree of over optimism.

[9] Individual rewards might nonetheless still arise in social contexts within the team (e.g., praise for work completed).

disputes, they should be willing to demonstrate some trust and confidence in the ability of the team to resolve its own issues and only intervene when appropriate (e.g., when social accountability measures are having no impact on improving team performance). For example, research has indicated that suboptimal performance can be both socially chastised within the group (e.g., open expressions of dissatisfaction within the group during a daily stand-up) or, in the event that it can be accounted for (e.g., unanticipated technical difficulty) may attract offers of support from other team members illustrating both the social and supportive nature of group dynamics. Working agreements may codify the terms of equitable behaviour by making clear what is considered acceptable input to the team (e.g., timeliness at meetings, adherence to coding standards).

There is, however, an organisational influence on motivation that concerns the degree to which a team can effectively operate in a motivated manner though it requires some unpacking to get to the bottom of it. Autonomy is the degree to which the execution of task offers freedom, independence and discretion in the scheduling of work and determination of how it is to be completed. This is influenced by many factors including organisational culture, rules and willingness to comply, levels of control reflecting expectations of management and the free will that an individual is willing to exercise. Closer inspection reveals that autonomy is manifest in multiple forms (i.e., organisation, group and individual) each of which interacts giving rise to tension:

- *External Autonomy*. This describes the influence of external stakeholders (esp. management) on team activities which can manifest itself in detrimental or beneficial outcomes. Examples include the setting of terms of references (e.g., corporate guidelines, contractual arrangements with suppliers, the assignment of team members to multiple projects) which can have the effect of creating a sense of powerlessness within the team, or diluting collective responsibility towards the project outcomes. Sometimes, however, external influence can be desirable as in the breakage of deadlocks or in the challenging of group think views held by the team.
- *Internal Autonomy*. This refers to the extent to which team members jointly share in decision-making authority and includes decision making processes. For example, the potential (mis)interpretation of the Team Leader or Project Manager roles in the DSDM as individuals who exercise exclusive decision-making authority on behalf of the group weakens internal autonomy considerably. Accordingly, decision-making models (e.g., delegation to subgroups, entire team involvement on particularly important matters) should be both appropriate and fluid for a healthy and robust internal autonomy to flourish.
- *Individual Autonomy*. This concerns the amount of freedom and discretion an individual possesses when performing their own tasks. A high degree of individual autonomy can erode team cohesion if it results in a reduction of communication

with other team members. The tension between individual and internal autonomy is referred to as the paradox of self-management [147]. This is often evident in an unwillingness to participate in meetings or a disinterest in communal events (e.g., daily stand-ups) since they appear unrelated to one's own activities.

If, as is widely cited in the agile community, motivation is the combination of mastery, purpose and autonomy[10] then it is necessary to appreciate that it is the right kind of autonomy that drives this motivation [204, 221]. In other words, significant imbalances between the various forms of autonomy may in fact stifle motivation it is perceived to detract from a common vision and purpose. Of these three mastery and autonomy are the most dominant elements since neither can result in motivation without the presence of the other as has been shown by empirical studies [64]. In other words, people only feel that they have competently achieved something when they have experienced self-determination in the process. Accordingly, micro-management of highly capable individuals leads down a path towards destruction of motivation.

The environment in with which an individual interacts also has a role to play since this may either impede or encourage motivational behaviours. For example, social-contextual events (e.g., feedback) may encourage feelings of competency and improve autonomy (e.g., words of encouragement or absence of demeaning assessments).

11.4 Commitment

Commitment can be defined as the relationship employees share with their organisation in terms of their beliefs in its goals and values, their willingness to exert effort (the level, direction and persistence of which are motivationally determined) on behalf of the organisation including their desire to remain within it [187]. Commitment, according to the three component model, represents a combination of one or more psychological states (i.e., feelings or beliefs concerning the relationship between employee and organisation) reflecting the desires, needs and obligations that an employee feels [174]. This model incorporates elements of both attitudinal[11] and behavioural[12] perspectives though more emphasis is placed on experience (of psychological states) than on attitudes themselves. Attitude in the organisational context refers to expressions of favour (or disfavour) towards the organisation, its people or events that occur therein and is thus a separate though not unrelated notion to commitment. The model, which has become popular amongst organisational psychologists

[10] The terms mastery, purpose and autonomy have become popular in the agile community though the original research from which they derive used the terms competence, relatedness and autonomy.

[11] The attitudinal perspective of commitment suggests that there are preconditions that determine a psychological state from which behaviours are derived.

[12] The behavioural perspective contends that behaviour as well as preconditions drive further (repeated patterns of) behaviour determine psychological states.

as a means for explaining why employees remain with their employers, defines the following three components of commitment:

- *Affective (desire)*. This is the form of commitment that arises through an emotional attachment to and an identification with the organisation and its values and goals and implies that membership of the organisation is a matter of choice.
- *Continuance (need)*. This commitment derives from balance of convenience considerations wherein the cost of leaving the organisation may be deemed too high (e.g., loss of benefits or social relationships).
- *Normative (obligation)*. This form of commitment is born of a sense of obligation towards the organisation (e.g., moral or social norms, expectations regarding duration of tenure).

It is important to note that each of the above is a component (not a level) of commitment and that an individual may experience one or more of these to a greater or lesser extent without exclusion of the others (e.g., continuance and normative commitment may both be present in an employee who is paid an above average salary and whose college tuition has been paid for by the company). Indeed, each component may be subject to variance over time and be affected by specific circumstances (e.g., continuance may grow over time as social relations begin to cement or affective commitment may be impaired by a corporate scandal). Factors that are known to have a negative impact on commitment include role conflicts and ambiguity (e.g., arising from incorrect alignment of personality and role, incompatibilities in function), lack of empowerment and support (e.g., inability to shape one's own environment, absence of action orientation, limited delegation of leadership) and non-reciprocity in working relations (e.g., job insecurity).

It therefore becomes evident that stable self-organised agile teams that enjoy a collaborative and empowering culture based on clear working agreements and roles ought to exhibit high levels of commitment towards the project and the organisation. In particular, there is some evidence that affective commitment, which leads to higher levels of initiative, productivity and quality [81], may be affected by organisational structures that are decentralised in terms of decision and policy making and that this is reinforced by support and role clarity, autonomy [180] and fairness and elements that contribute positively to self-identity (e.g., self-expression, personal importance). Though continuance commitment may be reinforced by the social bonds within stable teams and the perceived benefits of coaching and facilitative support, a genuine assessment of this form of commitment requires consideration of the alternatives (e.g., outside of the organisation) that is necessarily rather situational.[13] Finally, there is relatively little evidence concerning the sources of normative commitment that arises specifically in agile environments since this is often found to be more strongly related to the wider organisation or its policies (e.g., support of educational needs) that are difficult to tailor at the group level without attracting accusations concerning fairness and bias.

[13] The perceived distinctiveness (or lack thereof) of an organisation will likely feature in any comparison of the "costs" of leaving.

11.5 Emotional Intelligence

Emotional intelligence is defined as "a set of skills surrounding the correct appraisal and expression of emotion in oneself and in others, the effective regulation of emotion in self and others, and the use of feelings to motivate, plan and achieve in one's life" [94, 226] and is taken to usually taken to consist of the following elements [6, 97, 138, 139, 215, 280]:

- *Self-Awareness*. The ability to monitor feelings in real time which reveals personal insights and a deeper self-understanding. Central to this aspect is the attention to events without being carried away by emotions which requires being aware of our moods and our thoughts concerning those events. This is turn leads to better decision-making when one is attuned and attentive to one's own feelings.
- *Managing Emotions*. The appropriate handling of emotions that enables one to become more at ease and cope better with life's setbacks. This dealing with emotions includes the ability to observe, challenge or appraise and treat them. Note that the managing of an emotion (e.g., anger) requires that it be observed in the first place. Thus the managing of emotions builds on self-awareness.
- *Motivating Oneself*. The marshalling of emotions in the service of a goal (e.g., motivating oneself and being in command of impulsive behaviour) some of which are a reflection of specific emotional traits (e.g., enthusiasm and persistence) or practices (e.g., work ethic).
- *Recognising Emotions in Others*. The ability to identify and empathize with the circumstances of others (e.g., the "people diligence" of managers). This presumes an understanding of one's own feelings and includes the ability to read the signals given out by others (e.g., chiefly non-verbal cues such as tone, gesture, facial expression that capture how something is communicated). Empathy may also translate into specific forms and modes of action (e.g., ethical behaviour, altruism).
- *Handling Relationships*. The extent of social competence and interpersonal effectiveness. This requires self-control and calmness to be able to attune to the demands of others and the ability to understand the feelings of others in a manner that permits the further shaping of those feelings. Indeed it is believed that interpersonal intelligence comprises the ability to organise groups (e.g., initiating and coordinating efforts), negotiating solutions (e.g., mediation), personal connections (e.g., empathy) and social analysis (e.g., detecting and gaining insight in relation to feelings, motivations and concerns).

Self-awareness could be enhanced by those with a predisposition to experience intense emotional experiences which is one of the facets of positive Openness to Experience as described in the Five-Factor Model (see later in this chapter). High conscientiousness, and agreeableness together with low neuroticism, insofar as these include expressions of (high) competence, (high) altruism and (low) impulsiveness respectively, may signal an ability to motivate oneself and recognise emotions in others.

Emotional intelligence appears to be linked to flexibility (e.g., ability to cope with changing requirements) which is recognised as one of the key success factors in projects [149]. Given that autonomy is a driver of team flexibility, this suggests that emotional intelligence may have a multiplier effect on motivation within an agile team. Furthermore, emotional intelligence assumes strategic significance in light of relationship management required to build up anticipatory and reactive dynamic capabilities (see Chap. 2 for further details) which enables team members to access and integrate relevant knowledge into the project. This is of importance not only to lateral (e.g., Business or Technical Advisors), but also to vertical relations (e.g., Business Sponsor) where support may be required to buffer and protect the team.

11.6 Personality

As described in Chap. 4, the traditional profile of Solutions Developers as specialists (e.g., requirements, designer, system integrator) has given way to a collective ownership and responsibility that has promoted the notion of T-shaped skills sets. Behind this presumption of individuals with specific core skills that are complemented with wider business and technical skills, is the anticipation that such skills collectively cover the requirements of the project and that the necessary social skills are present to ensure team cohesion (e.g., tolerance, communication, conflict resolution). Thus whilst technical ability remains relevant, the extent to which it can be effectively deployed may well prove to rely to a large extent on personality traits. For example, team flexibility as measured in terms of autonomy and diversity together with the emotional intelligence of its members have been found to contribute positively to overall performance as measured in terms of product features, time to market and market success [97]. Furthermore there is clear evidence of the intensity of social relations in knowledge-based projects and the emotions that these elicit [116, 215] suggesting that an awareness of one's own emotions as well as those of other team members may directly impact performance. Over recent decades two approaches in particular, the Five-Factor Model and the Myers-Briggs Type Indicator, have emerged to become prominent arenas for the assessment of personality. Whilst little research has been done specifically in relation to Agile, that which has, has proven to be tantalisingly revealing.

11.6.1 Five-Factor Model (FFM)

The Five-Factor Model (FFM)[14] describes personality traits[15] in terms of a two-level hierarchy consisting of five factors each of which is comprised of six facets.

[14] The Five Factor Model is often referred to as the Big Five dimensions of personality though the origins of the former lie in personality questionnaires and that of the latter in lexical analysis.

[15] A trait is a feature that accounts for personality difference amongst individuals that is both stable and cross-situational.

Each factor was discovered through the statistical analysis of lexical trait terms corroborated by research with personality questionnaires. As an aside, the term *factor* arises from factor analysis, a statistical method that attempts to explain variability amongst a set of variables in terms of a reduced set of variables referred to as factors. Data observations are expressed mathematically as linear combinations of undetermined factors (together with an error term). Using regressional modelling, it is possible to locate statistically significant clusters which in turn leads to the identification of the factors. The following is a brief summary of each of the five factors (see their related facets in Appendix D for a more detailed description of the respective factors):

- *Openness to Experience*. This captures imagination and appreciation of and engagement in both artistic and intellectual[16] interests. Of all the factors, Openness to Experience has been the most controversial owing to differences in the lexical analysis and questionnaire approaches that gave birth to the FFM part of which is due to the absence of specific trait terms to describe aesthetic and artistic characteristics such as "sensitivity to art and beauty" [171].
- *Conscientiousness*. This trait values competence, orderliness and discipline and is associated with achievement-seeking behaviours.
- *Extraversion*. This dimension captures elements of sociability, gregariousness and assertiveness and is associated with positive emotions.
- *Agreeableness*. This factor embodies elements of trust, straightforwardness and tender-mindedness in relations with others.
- *Neuroticism*. This factor reflects the tendency to express negative emotions (e.g., anxiety, fear, depression) in response to pressure or stress.

The five factors have each been found to express a fundamental and enduring characteristic of personality that is independent of language and culture [33, 34, 169, 284]. Together these factors provide a relatively complete characterisation of a person at a global level reflecting the highest level of trait description. In practice, many observable characteristics draw into play one or more of these factors simultaneously (e.g., shyness might be accounted for by a combination of low extroversion together with elements of neuroticism) or be found to interact at the facet level (e.g., positive emotions, a facet of extraversion, might motivate sociability and activity which are facets of extraversion and agreeableness respectively). It therefore becomes important not to reduce someone to individual factors, but to consider the possible interactions between factors (e.g., a person who scores highly on Openness to Experience and Conscientiousness may express their inquisitiveness in systematic studies whereas had they scored low on Conscientiousness, this might have been expressed as idle curiosity [171]). Thus faced with the same broad range of issues, individuals exhibit different responses in terms of the way they think, act and feel which are reflected in broad terms by the factors. Whilst factors may be hereditary, there could also be environmental influences (e.g., underlying predispositions at the factor level whose

[16] The Intellect facet of Openness to Experience (see Appendix D "Five-Factor Model Facets") bears no relation to measured intelligence, i.e., it reflects personality rather than intellectual ability.

specific traits are expressed in certain environmental conditions). Although the FFM is by no means a full explanation of personality, it has withstood a wide range of criticisms [171] and shown itself to be empirically robust becoming one of the major strands of thinking in this field.

Although it is possible to determine the strength of each factor in an individual through the administration of a test (e.g., personality questionnaire) this approach seldom reveals significant insight at the factor level. For example, moderate Extraversion could reflect an individual who is either energetic, but aloof or lethargic, but friendly [170]. Accordingly, it is necessary to delve deeper into an assessment of the underlying facets in order to explain what is observed by the factors. In some cases, this is needed in order to understand the behaviour of an individual in a specific context (e.g., a moderate Openness for Experience that is accounted for by a pronounced intellectual rather than artistic interest might be of more interest in some agile settings).

Achieving a balance of personalities within a team is a challenge that requires careful consideration of what benefits the team as a collective. Essentially there are two matters to consider. First, there is the issue of what personality traits lend themselves to specific roles or functions within agile teams. Second, is the wider question of the overall compatibility of specific expressions of personality traits (e.g., high extraversion, low neuroticism) in relation to agile principles, practices and culture. Regarding the first issue, it should be noted that roles are more diffuse in agile teams (e.g., a developer might be involved in both architecture and testing) and that the techniques available to agile teams have greater social relevance.[17] Furthermore, whilst some factors incorporate traits such as activity and a willingness to assume control that might suggest leadership roles, it makes no assertion that such individuals will make *good* leaders. In addition factor level analysis can mask attributes in a manner that could misdirect decision-making (e.g., an average extraversion score of a good, but sober leader may be the same as that of a jovial individual who would rather that someone else took the lead). It is therefore to be expected that the second matter above, the wider compatibility of personality traits, plays a more significant role within the team (e.g., ability to resolve conflict, capacity to engage meaningfully in facilitated workshops, openness to innovative and new ideas). Though the research on what constitutes an optimal combination yields conflicting findings, there are still some general remarks that can be made in relation to agile environments [18, 182] that are illustrated in Fig. 11.3.

- Emotional stability (the opposite of Neuroticism) is conducive towards a cooperative environment and is known to contribute positively towards collaboration, though its precise impact on team performance remains somewhat unclear. For example, although cooperation and performance is improved in an emotionally stable team, this cohesion is sensitive to the presence of emotionally unstable individuals [20]. There also appears to be differing opinions on the extent to which

[17] One well researched technique, pair programming, is a good example of socio-technical practice though past studies of this technique have been rather inconclusive about the role that personality plays in team performance.

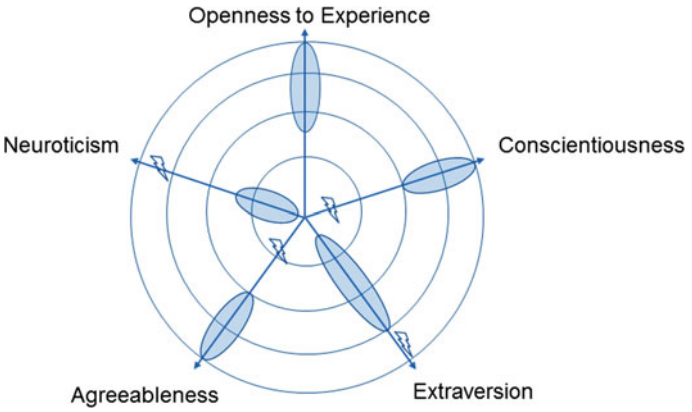

Fig. 11.3 Five Factor Model Applied to Agile Team Compositions. Published with kind permission of © Alan Moran 2015

variability of neuroticism within a team actually improves or impedes performance suggesting that the effects may be situational. In general terms though, high emotional stability is associated with performance in situations of relevance to Agile. Indeed within IT the evidence points to a positive contribution of emotional stability towards team performance among software developers and within small groups [69, 144].

- Extraversion may be necessary within a team in order to facilitate social interactions though getting the balance right appears to be important (i.e., too little extraversion stymies interactions and too much can give rise to conflict). For example, conflicts can arise if the Assertiveness facet is pronounced amongst those exhibiting extraversion.
- High levels of Agreeableness are particularly important in agile teams as they promote social interactions, direct the team towards common goals, foster openness and trust and contribute positively to cohesion and performance. However, such teams are vulnerable to the presence of even a single disagreeable individual who can significantly impact overall performance.
- Conscientiousness has repeatedly been shown to be associated with high performance, commitment towards goals and perseverance in getting there. Such individuals do not suffer fools gladly and the presence of individuals who score low in this dimension may be met with antagonism or even hostility. The impact of conscientious individuals can be felt largely independently of the project methodology (i.e., whether or not Agile is employed).
- Openness to Experience has been considerably less researched than the other factors in respect of team performance. Given the often complex and technical nature of agile projects, it might be assumed that the right kind of openness (e.g., strong expression of facets such as actions, ideas and perhaps values) may contribute positively towards performance.

For the agile manager therefore Agreeableness and Conscientiousness are highly desirable and a moderate range of Extraversion is to be welcomed as is good emotional stability (i.e., low Neuroticism). Team vulnerability in respect of particularly high Neuroticism or low Agreeableness needs to be actively managed and appropriate action taken before the team situation deteriorates (e.g., removal of an individual from the team). The skills that individuals possess provide a degree of restraint on what is achievable in terms of group personality profile. However, encouraging T-shirt shaped skill sets is likely to be a more appealing proposition to those with moderate to high Openness to Experience (provided that this is expressed in the appropriate facets) and will alleviate efforts to find an appropriate team composition. Furthermore, owing to the importance of team stability as a critical success factor of agile teams (see Chap. 4 for details of success factors), efforts should be made to keep a performant team together once a workable group profile has been established. Indeed, often the symptoms of instability (e.g., the need to swap out a specialist to other project teams) are merely a reflection of an inadequate distribution of skills that should be managed over the longer term (e.g., training, on-the-job learning) and this should accordingly be recognised as an entirely different management issue.

11.6.2 Myers-Briggs Type Indicator (MBTI)

The essence of the Myers-Briggs theory is that much seemingly random variation in behaviour is in fact rather orderly and consistent and due to basic differences in the way individuals prefer to use their perception and judgement [191]. Based on extrapolations of Jungian theory (in particular the existence of four basic mental models and the dominance of one of these in each individual) that were subsequently operationalized in the form of a type indicator, it is claimed that experiences are construed on the basis of preferences that determine interests, reactions, values, motivations and skills.

The basis of the Myers-Briggs Type Indicator (MBTI) is the fundamental orientation towards the objective or subjective world (extraversion versus introversion), the individual's relation to that world in terms of rational or irrational functions (judging versus perceiving) and the expression of one those as the dominant and the other as the auxiliary function (sensing versus intuition and thinking versus feeling). In spite of some inconsistencies in the original work of Jung [168], on which the MBTI is based, a consistent set of measures has emerged that enables classification into one of sixteen possible categories together with a numerical scoring of each type to clarify preference. This notwithstanding, there is still some evidence that these categories are not entirely independent of each other and that repeated takings of the MBTI can result in different type classifications over time though this is acknowledged amongst MBTI practitioners as type dynamics [102, 168, 205]. Moreover, the MBTI appears to be incomplete failing to capture the Neuroticism factor that was found to be fundamental in the FFM. That the MBTI is a typological (rather than a dimensional) approach has also not gone unnoticed or without criticism. For

example, whereas a dimensional approach measures a trait on a continuous scale (e.g., above average extravert), a typology makes a binary classification (e.g., extravert or introvert). There appears, however, to be little evidence of a bimodal distribution of preference scores [251]. Despite these reservations, the MBTI continues to enjoy widespread support and usage and classifies individual preferences in terms of the following four dichotomies:

- *Extraversion (E)/Introversion (I)*. This describes the attitude towards operation in the external world (e.g., action, people, objects) or the internal world (e.g., concepts, ideas) and is related to the notion of where an individual finds energy (e.g., through interactions and activity or in solitude and reflection).
- *Sensing (S)/Intuition (N)*. This dichotomy concerns how information is gathered indicating a preference towards tangible concrete data (i.e., sensing) or discovery of underlying principles and abstracted theories (i.e., intuition).
- *Thinking (T)/Feeling (F)*. Based on data gathered, this function is concerned with the manner in which rational decisions are made and whether there is a preference for logic and consistency (i.e., thinking) or for harmony and consensus (i.e., feeling).
- *Judging (J)/Perceiving (P)*. The final dimension, which represents an extension to the original model, indicates which preferred function prevails in relation to the external world based on either data gathering (i.e., perceiving) or data processing (i.e., judging). Thus individuals will either assume a dominant or an ancillary function (chosen from Sensing, Intuition, Thinking or Feeling) and employ their dominant function in their primary work (i.e., external or internal) and their auxiliary function in the other world.

Figure 11.4 illustrates the inherent ordering in the coding of MBTI characteristics. Beginning with where an individual prefers to get and direct energy, dominant information gathering and auxiliary decision making functions arise that are influenced by lifestyle orientation. The injection of the lifestyle interpretation (proposed in order to tackle perceived issues with the MBTI and establish a means of mapping it to the FFM [168]) introduced in light of the previous discussion of the FFM, at this point

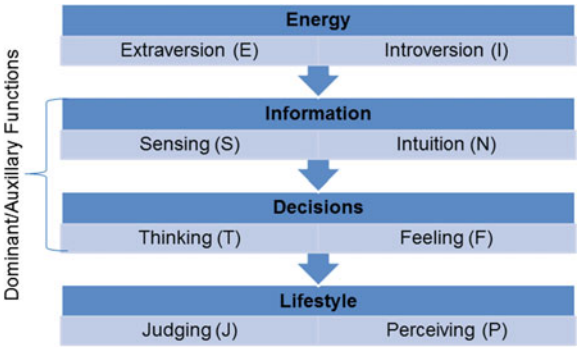

Fig. 11.4 The Myers-Briggs Personality Model (Adapted by Replacing Environment with Lifestyle). Published with kind permission of © Alan Moran 2015

Table 11.1 Distribution of MBTI Personality Types in the US Population [43]

ISTJ (11–14 %)	ISFJ (9–14 %)	INFJ (1–3 %)	INTJ (2–4 %)
ISTP (4–6 %)	ISFP (5–9 %)	INFP (4–5 %)	INTP (3–5 %)
ESTP (4–5 %)	ESFP (4–9 %)	ENFP (6–8 %)	ENTP (2–5 %)
ESTJ (8–12 %)	ESFJ (9–13 %)	ENFJ (2–5 %)	ENTJ (2–5 %)

is a direct challenge to the existence of dominant and auxiliary functions, but one that establishes a bridge between MBTI and FFM which will be returned to later.

Studies have repeatedly shown an over representation of certain personality types in specific occupations (e.g., higher density of intuitive individuals in writing and art). In part this is based on a presumed underlying distribution of types (see Table 11.1) though the stability and validity of this has been questioned [168]). This notwithstanding, amongst software engineers, there is a clustering of thinking preferences and a noticeable under representation of both sensing and feeling preferences [36, 156, 223, 246, 276]. The most common types are ISTJ (alone accounting for 25 % according to one study [14]) followed by INTJ and ENTP (which together make up 50 % of the personality profiles) though other similar studies have placed INTP in third place and found a higher proportion of individuals with S and N in IT settings in organisation that are innovative in nature [156]. Given the prevalence of thinking judging types and the moderate mix of extraversion and introversion, it would seem reasonable to expect an organisation with a strong IT focus to have a majority of xSTJ types. Indeed, one study found that together ISTJ and ESTJ types comprise approximately 65 % of the types amongst system analysts in an insurance organisation [246].

This points to the need to address the limitations of the most common personalities (e.g., capacity to grasp how tasks affect people and ability to understand user requirements from non-technical perspectives). The solution, however, is not to attempt to change personalities or remove individuals from the organisation, but rather to ensure that at the group level a sufficient mix of types exist that complement each other (e.g., add members to the team with the personality attributes that address any perceived deficiencies). For example, one study of pair programming practices demonstrated that MBTI diversity increased performance over control groups of identical and polar opposite types [44]. Diversity of type also appears to promote team cohesiveness though the results here are some more mixed [142].

As an aside, it would also appear that the positioning[18] of an organisation in the marketplace is important as this allows an organisation to better attract more appropriate individuals who are drawn to specific routine of work (e.g., solution development, delivery and maintenance versus product innovation, design and development). It might, for example, suggest that a company with a clear agile identification may

[18] Positioning in this context refers to the marketing notion of the association that springs to mind when one thinks of attributes of a company (e.g., those organisations that are perceived to be innovative).

Table 11.2 Mapping of FFM Factors to MBTI Dimensions

FFM factor	MBTI dimension	Comments
Openness to experience	Sensing-Intuition	Significant and very strong positive correlation
Conscientiousness	Judging-Perceiving	Significant and strong positive correlation
Extraversion	Extraversion-Introversion	Significant and very strong negative correlation
Agreeableness	Thinking-Feeling	Significant and strong positive correlation
Neuroticism		No correlation with any dimension

find a more diverse range of personalities amongst its job applicants than one that is associated with more conventional environments. The downside is that a significant realignment of corporate strategy together with related activities (e.g., mergers or acquisitions of companies of a very different character) may unsettle and disorient a workforce which up to that point may have expressed preferences entirely well aligned with the pre-existing corporate orientation.

11.6.3 Comparing FFM and MBTI

A detailed analysis of the relationship between FMM and MBTI [168] reveals that there is a correlation between the factors of FFM and the dimensions of MBTI though the latter does not accommodate Neuroticism.[19] Table 11.2 suggests that a mapping of FFM and MBTI traits might be possible permitting an MBTI assessment to translated into FFM terminology (e.g., an ENFJ is broadly speaking an extraverted, open, agreeable and conscientious individual). These findings are consistent across gender and rating systems (e.g., self versus peer) based on a sample of several hundred individuals over a wide age range [168].

The link between FFM and MBTI does not negate the need for or relevance of the other since the FFM is a purely descriptive framework and the MBTI attempts in part to explain underlying premises and potential mechanisms behind personality. The Jungian justification behind MBTI is, however, subject to criticism owing to lack of empirical data supporting the dominance of preferences and the use of the extraversion-introversion and judging-perceiving scales to determine dominant and auxiliary functions. The models tend to align rather better if the judging-perceiving

[19] The absence of Neuroticism in the MBTI is attributed in part to attitude of its authors to type only valuable and positive traits (though this fails to explain why Emotional Stability, the opposite of Neuroticism, was omitted) and the fact that this personality aspect does not feature in the underlying definitions derived from Jungian psychology.

dimension is interpreted in terms of a preference for structured or spontaneous lifestyles and flexibility is employed when borderline preferences are taken into account (e.g., when an assessment suggests that an individual may lie on the border of two or more MBTI categories). This notwithstanding, the mapping described in Table 11.2 enables access to the enormous body of knowledge accumulated during MBTI studies (e.g., density of specific personality types in certain occupations) allowing them to be interpreted in FFM terms. The best advice, therefore, when using MBTI might well be to ignore its Jungian origins and interpret any findings in the language of the FFM.

11.6.4 Preference for Agile

Given that certain personality types are disproportionally represented in specific occupations, it is valid to ask whether or not there is a personality preference towards working in agile environments [40–42]. Surprisingly there has been little research conducted in this area and that which has been largely inconclusive because these results are rather tentative, sensitive to sample size and some of the methodological practices are questionable [26, 27]. Sadly, however, several core principles (e.g., frequent delivery of software) are eliminated from this analysis in the mistaken belief that they represent universally desirable characteristics and fail to distinguish agile from traditional methods (this is particular unfortunate in light of analysis that shows them to be coherent software engineering principles [239]). Elsewhere it is unclear if the FFM factors are correctly aligned with agile culture (e.g., Openness to Experience is said to be aligned with "action oriented posture" although activity is a facet of Extraversion). Finally, attempting to attribute variance in agile preference *entirely* to personality is perhaps unreasonable so other factors are posited to account for variations in preference (ironically these include delivery of software as a feedback mechanism). What does, emerge, however, is the general impression that Extraversion (esp. social awareness), low Neuroticism and moderate levels of Openness to Experience all indicate a preference towards Agile. Though one might expect Agreeableness to correlate positively it would seem that group reflection and the dynamics of self-organisation require some introspection (e.g., group reflection) and a willingness to engage in a degree of discord and dispute in order to resolve issues within the group [27].

11.6.5 Ethical Considerations

Ethics, concerned with values relating to human conduct and concepts of right and wrong, is an important consideration when applying psychological models in the workplace. Personality is a complex and subtle topic and despite significant advances in research over the past decades, understanding of individual and groups behav-

iours together with how traits relate to group dynamics and performance remains incomplete. Furthermore, assessment instruments vary and may offer inadequate explanations of psychological profiles (e.g., FFM assessment at factor rather than facet level). Indeed, the Myers-Briggs Foundation specifically states that the use of its test to make work assignment decisions or to provide career guidance is unethical stressing that its tests identify preferences and do not imply competence, ability or excellence nor should they be used to stereotype or label individuals (e.g., screening job applicants). Similarly the adequacy of the Five-Factor Model for employment selection purposes (and by implication role assignment for incumbent employees), has also been called into question [230, 247]. On the other hand, psychological tests administered to willing participants whose results are not communicated, knowingly or unknowingly, to third parties are perfectly acceptable. Indeed, they provide a valuable opportunity for the individual to learn more about themselves including their preferences, inclinations and development needs.

11.7 Management Implications

The agile principle that teams should be built around motivated individuals (see Appendix A) appears rather optimistic given that most teams will contain individuals whose motivation is not entirely aligned with the project objectives. This implies that motivation is a management concern that must be addressed appropriately. Autonomy, team diversity and flexibility all impinge on motivation and commitment which together demonstratively interact leading to enhanced performance wherein emotional intelligence (e.g., management) provide a guiding hand. In fact, studies have found that emotional intelligence facilitates exchange of information and decision-making which in turn enhances team performance. These findings are attributable to a study [97], that looks at a specific subset of emotional intelligence covering recognition (i.e., ability to perceive emotions and understand their potential causes and effects) and regulation (i.e., ability to manage one's own emotional expressions and those of others) and their impact on performance in software development projects. The implications and challenges for management therefore can be summarized as follows:

- Autonomy, a driver behind team flexibility and a key element of motivation, should be nurtured and fostered in order to enhance it. Concrete measures might include permitting the team to select the most appropriate tools and technologies and empowering them to determine what could be achieved by the project within the broad guidelines laid down by more senior management. Therefore the challenge is to ensure that appropriate controls are in place that prevent the team descending into chaos and instability whilst balancing these against rigidity that would otherwise stifle or impair creativity.
- Determine the optimal mix of autonomy (i.e., external, internal and individual) in order to foster affective commitment and team performance. This might take

the form of protecting the team from undesirable external influences (i.e., external autonomy), ensuring appropriate decision-making models are in place within the team (i.e., internal autonomy) and making sure that individuals continue to relate their work with that of the rest of the team (i.e., individual autonomy).

- Team diversity should be promoted by ensuring variety in terms of specialisms, background cultures and perspectives. This not only ensures that the team as a collective possesses the necessary skills to achieve its goals, but also increases the possibility of identifying creative solutions in the problem solving process. Thus avoid highly specialised skills resulting in a division of work that reduces flexibility and internal autonomy. It also helps to balance out individual shortcomings from a personality perspective and builds up of redundancy through information exchange and sharing of skills. Indeed, redundancy could be said to be an important prerequisite for self-organisation and autonomy within teams [180, 256].
- Convey the importance that empathy plays in improving team performance (e.g., as measured by delivery in shorter timeframes of products that are more functional). This requires balance of views and diversity of opinion within the team and the emotional ability to engage and identify with the needs of the customer. It may help to increase awareness that individuals bring not only skills and knowledge, but also feelings and emotions into the team and that these can be expressed to the benefit of the team and the achievement of its objectives [143]. This necessitates the creation of a safe environment in which trust can flourish enabling exchange of experiences and opinions and the expression of feelings and emotions.

Improving motivation requires a focus on enhancing mastery and autonomy and ensuring that these are matched (e.g., placing a low skilled individual in a highly autonomous environment is detrimental to motivation). Individuals are encouraged to initiate the journey to improved motivation by that which they perceive as being valued or promoted by those whose respect they hold. This means that leadership must imbue actions with meaning and act with integrity. Moreover, they agile managers must facilitate this journey through encouragement and support and be mindful of practices that may reverse motivational trends (e.g., use of external regulation or introjection techniques). For example, individual financial bonuses are considered a regressive measure when compared with the other tools available to a manager (e.g., promotion of self-endorsement of tasks or goal congruence, being autonomy supportive). The focus must therefore be on identification and integration (see Fig. 11.2) and the use of mastery and purpose to promote the former and autonomy to enhance the latter. Indeed, it is autonomy that permits an individual to integrate values into their own belief system and thus without it, management are actively inhibiting motivation.

Turning attention to personality, the prevalence of xSTJ types reported by MBTI studies in IT environments correlates positively with open, agreeable and conscientious individuals as suggested by the mapping to FFM factors (see Table 11.2). This hints at good preferential compatibility with agile approaches as indicated by Fig. 11.3 despite the lack of a more detailed facet level explanation. The slight leaning towards introversion amongst software developers and related roles

(e.g., systems analysts) appears to be no cause for concern as an optimal mix requires both introversion and extroversion neither of which is as directly related to social interaction as common stereotypes would suggest. More relevant is addressing the potential limitations of some common personalities such as the capacity to grasp how tasks affect people and ability to understand user requirements, may benefit from wider team diversity. That diversity is a strength would seem particularly to be the case when a team is faced with challenging projects that require high quality, creative and innovative solutions [165] where apparently agile diversity thrives. From a management perspective MBTI highlights the need for people skills amongst managers in an organisation and suggests that greater diversity within agile teams may be beneficial to overall team performance. Managers operating in high velocity environments themselves require a degree of emotional intelligence in order to cope with uncertainty in particular when it comes to strategic decision-making [76].

As indicated already agile managers need also to be aware of the potential detrimental impact of some traits (e.g., high neuroticism, low conscientiousness or agreeableness) and be prepare to take appropriate action. Moreover, given that some elements of trust and cooperation that are important in agile cultures are facets of Agreeableness not accounted for in the Thinking-Feeling type and that other factors (e.g., Neuroticism) do not feature at all in MBTI, caution concerning an over reliance on typologies is to be warranted. Given the apparent widespread application of MBTI in Human Resources (in spite of ethical concerns expressed by the methodology) this suggests that some decision-making in relation to Agile could be misplaced or ought at least to be challenged.

Appendix A
Agile Manifesto and Principles

A.1 Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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

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

This statement is subject to copyright according to the terms laid out in [24].

A.2 Agile Principles

We follow these principles:

- 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 to 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.

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1

- 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 behavior accordingly.

Appendix B
Agile Techniques

The following list of agile techniques reflects those practices (and their interpretation) most commonly encountered by the methodologies in this book. The list should not be understood as exhaustive and we refer the interested reader to the primary sources for more detailed descriptions and advice.

Technique	Description
Agile Charting	Cyclical representation of agile processes used to communicate intent and solicit feedback within teams
Acceptance Test	Formal and often technically verifiable statement of what constitutes done in the form of a test that returns either success or failure. This enables clarification of what is to be expected of a solution and is related to unit testing (which is a form of white box testing that occurs at a lower level) and automated practices such as test-driven development
Behaviour-Driven Development	Focus of testing on behavioural aspects that contribute most to business value. Considered a refinement of Test-Driven Development it bears several different nuances when specifying user stories (e.g., phrases them in given-when-then terms, drilling down to their core purpose using the Kaizen style five why technique)
Burndown Charting	Regularly updated display of project progress (usually in terms of decreasing remaining effort)
Coding Standards	Adherence to a common standard (perhaps verified by static and runtime code analysis) throughout the codebase
Continuous Integration	Continuous merging (and typically unit testing) of code contributions into a shared repository in order to reduce integration hurdles and provide feedback at the code commit level. The principles and practices of continuous integration are detailed in [74]
Daily Build	The practice of building and packaging the solution on a daily basis in order to ascertain its technical deployment readiness. Whilst such builds are often continuously integrated, this practice refers to the making available of the emergent solution in such a form that stakeholders can interact with it and provide timely and meaningful feedback

(continued)

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1

Technique	Description
Daily Standup	Brief and focused daily meeting of all team members focusing on what has been achieved since the last meeting, what is planned that day and what is blocking work
Definition of Done	Criteria list of what must be achieved in order to deliver an increment though expression may find itself at the story or iteration level. This is closely related to the Definition of Ready concept which concerns the acceptance into an iteration of user stories based on verifiable criteria
Demonstration	Presentation of the state of an evolving solution at the end of an iteration in order to solicit feedback from stakeholders
Facilitated Workshop	Structured workshop wherein team members work towards specific goals with the support of an independent facilitator
Heterogeneous Small Teams	Formation of teams, with preferably five to nine (this oft cited team size number owes its heritage to [176]) co-located members comprising of business representatives and solution developers, with a preference towards generalists over specialists
Incremental Delivery	Release into production environments of partial evolving solutions throughout the project lifecycle
Information Radiator	A dashboard concept presenting project relevant information at a location to which project members naturally feel drawn (e.g., coffee machine)
Kanban	Pull based taskboard (with status swimlanes) where team members assign tasks to themselves and use states to update progress (e.g., “In Progress”, “Completed”)
Refactoring	The restructuring of code internals in a manner that does not alter its outward behaviour. Refactoring was first described in detail in [83]
Increment Planning	Creation of high level (and often imprecise) plans based on available knowledge
Modelling	Creation of a conceptual design (e.g., process diagram) for the purpose of discussion and gaining of consensus concerning a problem or its solution
Iteration Planning	Creation of detailed short term plans (typically based on user stories) complete with estimates and priorities and derived from the increment plan
MoSCoW Prioritisation	Classification of priorities in terms of MUST, SHOULD, COULD and WONT (in respect of the current timeframe!) and the accommodation of contingency in planning
Pair Programming	Working together of two solution developers on the same piece of code at the same terminal often interchanging the roles of implementer and reviewer
Planning Poker	A variant of the Wideband Delphi method in which consensus is sought during prioritisation by requiring participants to vote on and discuss their priorities
Product Vision	Formulation of a simple evocative statement that clearly states the purpose of the product which the project is endeavouring to create
Prototyping	Creation of a (possibly throwaway) mock to trial the design of a solution or to better understand the problem
Retrospective	Gathering of all team members to discuss lessons learned during the last iteration with a view to identify and act on possible process improvements. The definitive guide to agile retrospectives is [66]

(continued)

Technique	Description
Sustainable Pace	Maintenance of normal working hours and the avoidance of fluctuations and peaks (e.g., as project deadlines approach)
Test-Driven Development	Creation of unit tests prior to developing solutions as part of a culture of continual validation and verification. Test-Driven Development was first elaborated on in [23]
Timeboxing	Structured interval of time to segment and control activities
User Stories	Formulation in simple narrative form of requirements together with statements of what constitutes successful implementation (definitions of done). Typically this is expressed in a statement of the form "as a *role* ... I want to *objective* ... so that *justification*"
Iterative Development	Traversal of the entire SDLC within a fixed timeframe with the aim of producing tested and stable code that contributes towards the evolving solution

Appendix C
DSDM Roles

C.1 DSDM Agile Project Management Framework Roles

C.1.1 Project Level Roles

See (Table C.1).

Table C.1 Agile project framework project level roles

Role	Description
Business Sponsor	This role, usually assumed by a senior business individual acting in an executive capacity and with a strategic focus, is responsible for the formulation of the business case and will assume ownership of the solution once it is delivered. Key is the ownership of budget and accountability for return on investments (i.e., the right project for the right price). This role provides funding, ensures effective decision making and is the final point for escalation. In large organisations, this role may be split amongst several individuals though in such cases a "first amongst equals" ought to be determined in order to liase as a single point of contact with the project
Business Visionary	This role sets and promotes the business vision by directing project activities towards a solution that enables the benefits outlined in the business case and is thus astutely aware of the organisational and operational impact of the project on the organisation. Defining elements of this role include responsibility for alignment of requirements with business vision, delivery of business change, ensuring that incremental delivery thereof generates optimal value and overall fitness for purpose. This role approves all changes to the prioritised requirements list (incl. setting of key requirements), maintains a watch on business related risks, secures business resources and ensures collabortion with business stakeholders and is the final arbiter in team disputes. Unlike the sponsor, this role ought to be assigned to a single individual

(continued)

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1

Table C.1 (continued)

Role	Description
Project Manager	This role is concerned with the day to day management of the solution development environment and supports the planning undertaken by the team lead and the solution development team. In essence this role is responsible for delivery of project products within budget and subject to constraints and must manage, control and track in order to ensure delivery of a fit for purpose outcome. The project manager assumes responsibility for ensuring effective use of funds provided to deliver the solution within agreed timescales, monitoring of progress, risk management, resourcing of specialists as needed, communication to governance authorities, together with a variety of soft duties (e.g., conflict resolution, motivation, coaching). The project manager should not, however, impede empowerment within the solution team and is therefore not responsible for detailed planning, task assignment or related command and control functions
Technical Co-ordinator	The technical coordination role is the technical equivalent of the Business Visionary and is responsible for technical design along with consistency and quality of solution development. This role must ensure the integrity and alignment of the proposed solution design and its objectives (incl. compliance to standards and technical fitness for purpose). Specific duties may include identification and ownership of the technical architecture and environment (incl. identification of risks), co-ordination of technical activities and the transition of the solution into production, ensuring compliance in relation to standards, providing advice and assitance in matters of non-functional requirements and resolution of technical differences. Prior to its removal with the advent of the DSDM Agile Project Framework, the Technical Co-ordinator also assumed responsibility for configuration management

C.1.2 Solution Development Team Roles

See (Table C.2).

Table C.2 Agile project framework solution development roles

Role	Description
Team Leader	Playing a servant-leadership role, the Team Leader ensures full participation of team members wihin their roles and empowerment scope, ensures that iterative development and testing activities are conducted, assists in bring identified risks to the attention of the project level roles together with a variety of facilitation activities (e.g., daily stand-up meetings, reviews and retrospectives). There can be fluid leadership character to this role which may be held by different (ideally elected) individuals throughout the project
Business Ambas-sador	This role is selected from the business front line and has the capacity to make decisions in the interests of the users of the solution. Whilst key high level requirements are set out by the Business Visionary, this role is concerned with the detailed requirements to be implemented at the timebox level. It lies with this role to manage expectations, assess the evolving fitness for purpose and to validate deliverables during testing (e.g., acceptance testing, providing realistic test scenarios) together with the Solution Tester who verifies the solution and ensuring training for those using the solution

(continued)

Table C.2 (continued)

Role	Description
Business Analyst	This role is focused on the interface between business and technical domains and ensures the consistency of the final solution in terms of how it services business needs. In so doing, the Business Analyst ensures alignment of objectives and proposed solution (an activity shared with the Technical Co-ordinator) and may collaborate with others to create the Business Case. This role entails analysis of functional and non-functional requirements in the context of the identification of solution options (e.g., modelling, risk identification), facilitating communication between business and technical partners and assessing the impact, fitness for purpose and viability of solutions. An individual holding this role is likely to possess the analytical skills required to capture business processes and requirements (e.g., requirements engineering) and propose new designs for business processes and organisational change. They should also be in a position to assess the extent to which the proposed benefits have indeed been realised
Solution Developer	This role translates business functional and non-functional requirements requirements into a technical solution enuring in the process that standards are ahdered to and that outputs are tested. A variety of other activities also fall under daily development scope (e.g., modelling, quality assurance, reworking of solution components in the light of new insights and changes)
Solution Tester	The Solution Tester performs testing throughout the project and is primarily concerned with verification of the solution. This role works together with the Business Ambassador and Business Analyst who collectively are responsibility for validation of the solution. Specific responsibilities include management and execution of test scenarios and test cases, identification of testable acceptance criteria, communication of test progress and ensuring adequate test coverage

C.1.3 Supporting Roles

See (Table C.3).

Table C.3 Agile project framework ancillary roles

Role	Description
Business Advisor	It is likely that this role will be a subject matter expert possessing highly specialist knowledge that will be called upon by the Business Ambassador to advise or assist in project decisions and activities. This role may engage in a wide variety of ways (e.g., providing advice, consultancy or documentation, assessing or devising solutions and tests)
Technical Advisor	This role plays a similar role to the Business Advisor, albeit from a technical perspective. The role may contribute in a variety of forms such as specialist technical advice, operational insights, quality assurance or technical testing
DSDM Coach	This role assists the team at a methodological level by advising and supporting them in undestanding of the methodology, resolution of differences and conflicts regarding project approach. As required, the Coach may influence the tailoring of the process to project specific circumstances. This role requires a profound and deep understanding of DSDM as well as how best to adapt it

C.1.4 Workshop Roles

See (Table C.4).

Table C.4 Workshop roles

Role	Description
Workshop Owner	The owner funds the workshop event and sets the topic, parameters, objectives and deliverables. Depending on scope, the Workshop Owner role is usually assumed by the Business Sponsor, Business Ambassador or Team Leader
Workshop Facilitator	The facilitator is responsible for assisting the group disucssion process and ensuring that objectives are and that a safe environment for discussion exists and is maintained. The facilitator ensures that the topic is clear and sufficiently well defined, establishes the agenda, defines the workshop format and organises the event (e.g., booking of faciltities, sending invites to the participants). A facilitator requires good communication and listening skills, the ability to move a discussion forwards, knowledge of when it is appropriate to intervene, tact and empathy with the ability to build trust in the group. Of critical importance is the neutrality and independence of the facilitator who must not be seen to be steering the group towards a preferred or predetermined outcome. Therefore, the facilitator must remain focused on the process and not the content even when (s)he has views or opinions concerning the topic under discussion. Some agile commentators recommend that briefly stepping out of the role of facilitator might be acceptable if the facilitator has a view on the content. This, however, should be understood as a facilitation issue which might reflect a conflict of interest or be perceived as a source of bias. In such cases, it is always preferable to have used an independent faciltiator chosen from outside the project stakeholder group
Workshop Scribe	This optional role records the outputs of the workshop (e.g., objectives met, decisions made, unresolved issues) and may support in the creation of models and diagrams to support the group discussion
Workshop Participant	The individuals taking part in the content of the disucssion must possess the necessary skills and experience to be able to meaningfully contribute
Workshop Observer	An observer may be present for reasons unrelated to the content or facilitation of a workshop (e.g., auditor, trainee). Whilst in theory an observer neither takes part in nor influences the discussion, there mere presence can have an effect on the group which the facilitator must take account of and act accordingly

C.2 DSDM Agile Programme Management Framework Roles

C.2.1 Programme Management Team

See (Table C.5).

Table C.5 Agile programme management framework programme management team roles

Role	Description
Business Programme Owner	This role represents the senior executive support for the programme and the commitment to the realisation of its capabilities and benefits. It must provide strategic oversight, be in command of the necessary resources and funding, ensure alignment with business strategy, maintain relations with key stakeholders and be the final arbitrartor of escalations and high-level issues
Business Change Owner	This role has remit over a specific business area and ensures the realisation of benefits in that area without jeopordising the overall consistency of capabilities. At the project level this role could be identified with the Business Sponsor who works on details together with Business Change Agents. Thus alignment of benefits and projects and ensuring that this remains so in the light of changes in the wider business environment together with management of business area stakeholders and responding to project level escalations all lie in the responsibility of this role
Programme Manager	Delivery of the programme (incl. capabilities and benefits realisation mechanisms) come under this role who must work together with project teams and Business Change Owners to enable them to perform detailed planning and ensure that benefits are realised respectively. Similar to Business Change Owner the Programme Manager's duties overlap with those of the Business Sponsor at the project level if the tension between pursuit of project and programme levels goals is to be resolved. A variety of duties are embodied in this role including programme planning and risk management, dealing with stakeholders and managing intra-project dependencies and conflicts

C.2.2 Capability Delivery Team

See (Table C.6).

Table C.6 Agile programme management framework capability delivery team roles

Role	Description
Programme Technical Architect	This role ensures programme level consistency of technical details and architecture. Similar to the project level Technical Co-ordinator duties generally revolve around adoption of best practices and standards, resolution of intra-project technical design issues (e.g., interfaces, dependencies) and monitoring and controlling technical architectures and environments
Programme Change Architect	Linked to the Business Architecture Model, this role ensures its consistency and clarity of understanding across the projects and adapts and changes the model where appropriate
Stakeholder Engagement Co-ordinator	This role is engaged with stakeholders at all levels ensuring that they are informed and on-board with the programme. This requires considerable communication and diplomacy ensuring that stakeholders understand their roles and remain committed to them
Project Teams	All roles cited at the project level contribute directly to capability delivery and are subsumed at this level in the programme role hierarchy

C.2.3 Supporting Roles

See (Table C.7).

Table C.7 Agile programme management framework supporting roles

Role	Description
Specialists	Inevitably there will be on-demand need for the involvement of specialists in the programme and this catch-all role encompasses their involvement in facilitating and supporting the programme
Change Agent	Frequently the enabling of a capability requires change to existing processes and organisational culture before its benefits can be realised. In such cases, the Business Change Owner may engage one or more Change Agents to implement such changes
Programme Support Officers	The Programme Support Office comprises of individuals who ensure the smooth operation of the programme by providing administrative support (e.g., templates, handling of documents) thoughout. Note that these roles do not assume managerial responsibility which resides primarily at the Programme Management Team level

Appendix D
Five-Factor Model Facets

The Five-Factor Model (FFM), introduced in Chap. 11, describes a two-level hierarchy of factors and their related facets which are listed for reference (Table D.1).

Table D.1 Factors and facets of the five-factor model

Factor	Facet	High descriptors	Low descriptors
Extraversion	Warmth	Socialable with genuine affinity towards others and open demonstration of positive feelings towards them	Unlikely to reach out to others and seen as distant or reserved and perhaps difficult to get to know
	Gregariousness	Find reward and stimulation in the company of others	Overwhelmed by the presence of a crowd and prefers to be alone
	Assertiveness	Talkative and assuming of leadership roles, taking charge of and directing others	Quiet and willing to let other control the activities of the group
	Activity	Energetic and vigorous lifestyle with diverse range of activities	Leisurely, even sedate and relaxed, pace of life
	Excitement-Seeking	Need for high levels of stimulation and comfort with risk taking and thrill-seeking	Overwhelmed by commotion (e.g., loud noise) and aversion to thrill-seeking
	Positive Emotions	Positive mood and feelings including happiness, enthusiasm and optimism	Less prone to high spirits and strong emotions
Agreeableness	Trust	Presumption of fairness, honesty and well-intendedness in others	Propensity to regard others as selfish, devious and even dangerous

(continued)

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1

Table D.1 (continued)

Factor	Facet	High descriptors	Low descriptors
	Compliance	Candid, frank and sincere in dealings with others and unlikely to cheat	Guarded approach with others with resistance towards complete openness and may employ flattery
	Altruism	Willing to assist those in need in which a sense of self-fulfilment is achieved	May consider requests for help as intrusive and be perceived to look down at others
	Straightforwardness	Distaste for confrontation and willingness to compromise even to the extent of denying own needs	May have a sharp tongue and be willing to employ intimidation to get own way
	Modesty	Dislike for being the centre of attention	May think highly of oneself bordering on arrogance
	Tender mindedness	Display of compassion and sympathy and may be moved by the pain of others	Not strongly affected by human suffering and may possess a strong sense of rational justice
Conscientiousness	Competence	Strong confidence in ability to accomplish things together with drive and self-control	Pervasive sense of not being in control of one's own life and prone to misjudgement
	Order	Well-organised life governed by routines and schedules managed by lists and plans	Perception of being disorganised and scattered
	Dutifulness	Strong sense of moral obligation and willingness to follow the rules	Sense of being restrained by rules and regulations and may be perceived as unreliable or irresponsible
	Achievement-striving	Striving for excellence and a desire to be acknowledged and recognised	Satisfied doing just enough to get by and may be perceived as being lazy
	Self-discipline	Ability to commit sustained effort and show perseverance when completing tasks	Propensity to procrastinate or allow oneself to be distracted when trying to complete tasks
	Deliberation	Cautiousness and consideration ahead of engaging in action	Lack of consideration of alternatives and their consequences when undertaking tasks

(continued)

Table D.1 (continued)

Factor	Facet	High descriptors	Low descriptors
Neuroticism	Anxiety	High anticipation of danger reflected in fear, nervousness and tension	Overriding impression of calm and fearlessness
	Hostility	Resentfulness and bitterness when subjected to perceived unfairness and feelings of anger when things do awry	Impassive composure indicative of someone who is unlikely to become irritated or irate
	Depression	Inhibition towards initiating of activity driven by sense of sadness and dejectedness	Strong sense of comfort with oneself
	Self-consciousness	Heightened sensitivity concerning what others think of oneself and fear of rejection or humiliation	Unlikely to feel nervous in social settings and less likely to assume that one is being judged by others
	Impulsiveness	Difficulty in coping with strong cravings and urges and perhaps prone to binging or hedonistic activities	Seldom tempted to overindulge and not generally subject to cravings
	Vulnerability	Feelings of panic, confusion and helplessness in situations of stress or pressure	Capable of clear and poised thinking when in stress
Openness (to Experience)	Fantasy	Expressive and vivid imagination (e.g., day dreaming)	Finds comfort in facts over fantasy
	Aesthetics	Appreciation of beauty in art and nature and a perception of value therein	Lack sensitivity and interest in artistic matters
	Feelings	Good understanding of and awareness of own feelings born of intense emotional experiences	Less aware of own feelings and difficulty in articulating and expressing emotions
	Actions	Counter the boredom of routine by engaging in new experiences	Reluctance to embrace change and comfort found in routine
	Ideas	Enjoyment in discovering and debating new ideas	Preference of people over ideas and a tendency to discount intellectual exercises
	Values	Readiness to challenge authority and conventions and a capacity to cope with ambiguity	Strong belief in security and stability grounded in traditional values

References

1. Nelson, C., Taran, G., de Lascurain Hinojosa, L.: Explicit risk management in agile processes. In: Agile Processes in Software Engineering and Extreme Programming, pp. 190–201. Springer, Berlin (2008)
2. Abrahamsson, P., Warsta, J., Siponen, M., Ronkainen, J.: New directions on agile methods: a comparative analysis. In: 25th International Conference on Software Engineering, pp. 244–254. (2003)
3. Adams, J.: Risk. UCL Press, London (1995)
4. Ahola, T., Ruuska, I., Artto, K., Kujala, J.: What is project governance and what are its origins? Int. J. Proj. Manag. **27**, 131–141 (2013)
5. AIRMIC, ALARM, IRM: A risk management standard. AIRMIC, ALARM, IRM (2002)
6. Akgün, A., Keskin, H., Byrne, J., Günsel, A.: Antecedents and results of emotional capability in software development project teams. J. Prod. Innov. Manag. **28**(6), 957–973 (2011)
7. Alderfer, C.: An empirical test of a new theory of human needs. Organ. Behav. Hum. Perform. **4**(2), 142–175 (1969)
8. Ambler, S.: Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. Wiley, New York (2002)
9. Ambler, S.: Generalizing specialists: improving your it career skills. http://www.agilemodeling.com/essays/generalizingSpecialists.htm (2003)
10. Ambler, S.: Quality in an agile world. Softw. Qual. Prof. **7**(4), 34 (2005)
11. APM: A guide to governance of project management. Association for Project Management (2004)
12. Arnuphaptrairong, T.: Top ten lists of software project risks: evidence from the literature survey. In: Proceedings of the International MultiConference of Engineers and Computer Scientists I, pp. 16–19. (2011)
13. Artto, K., Kujala, J.: Project business as a research field. Int. J. Manag. Proj. Bus. **1**(4), 469–497 (2008)
14. AS/NZS: Australian and New Zealand risk management standard AS/NZS 4360:2004. Standards Australia (2004)
15. Atrill, P., McLaney, E.: Accounting and Finance for Non-specialists. Pearson Education, Harlow (2006)
16. Bach, J.: Exploratory testing explained. http://www.satisfice.com/articles/et-article.pdf (2003)
17. Baker, J.: Tightening the iron cage: concertive control in self-managing teams. Adm. Sci. Q. **38**(3), 408–437 (1993)

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1

18. Balijepally, V., Mahapatra, R., Nerur, S.: Assessing personality profiles of software developers in agile development teams. Commun. Assoc. Inf. Syst. **18**(4), 55–75 (2006)
19. Balkema, A., Molleman, E.: Barriers to the development of self-organizing teams. J. Manag. Psychol. **14**(2), 134–150 (1999)
20. Barrick, M.: Relating member ability and personality to work-team processes and team effectiveness. J. Appl. Psychol. **83**(3), 377–391 (1998)
21. Baskerville, R., Levine, L., Pries-Heje, J., Balasubramaniam, R., Slaughter, S.: How internet software companies negotiate quality. Computer **34**(5), 51–57 (2001)
22. Batra, D., VanderMeer, D., Dutta, K.: Extending agile principles to larger, dynamic software projects: a theoretical assessment. J. Database Manag. **22**(4), 73–92 (2011)
23. Beck, K.: Test-driven Development: By Example. Addison-Wesley Professional, Boston (2003)
24. Beck, K.: Manifesto for agile software development. http://agilemanifesto.org/ (2001). Accessed 1 Oct 2014
25. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change. Addison-Wesley, Reading (2004)
26. Bishop, D.: Personality theory as a predictor for agile preference. In: Proceedings of the Eight Midwest Association for Information Systems Conference (2013)
27. Bishop, D., Deokar, A.: Toward an understanding of preference for agile software development methods from a personality theory perspective. In: IEEE 47th Hawaii International Conference on System Sciences (HICSS), pp. 4749–4758. (2014)
28. Black, R.: Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing. Wiley, Hoboken (2009)
29. Boehm, B.: A spiral model of software development and enhancement. ACM SIGSOFT Softw. Eng. Notes **11**, 14–24 (1986)
30. Boehm, B.: A spiral model of software development and enhancement. IEEE Comput. **21**, 61–72 (1988)
31. Boehm, B.: Software risk management: principles and practices. IEEE Softw. **8**, 32–41 (1991)
32. Boehm, B., Turner, R.: Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley, Boston (2009)
33. Bond, M., Nakazato, H., Shiraishi, D.: Universality and distinctiveness in dimensions of Japanese person perception. J. Cross Cult. Psychol. **6**, 346–357 (1975)
34. Borkenau, P., Ostendorf, F.: Untersuchungen zum Fünf-Faktoren-Modell der Persönlichkeit und seiner diagnostischen Erfassung. Zeitschrift für Differentielle und Diagnostische Psychologie **10**, 239–251 (1989)
35. Brown, J., Duguid, P.: Organizational learning and communities-of-practice: toward a unified view of working, learning, and innovation. Organ. Sci. **2**(1), 40–57 (1991)
36. Buie, E.: Psychological type and job satisfaction in scientific computer professionals. J. Psychol. Type **15**, 50–53 (1988)
37. Burgelman, R.: A process model of strategic business exit. Strateg. Manag. J. **17**, 193–214 (1996)
38. Byrd, T., Turner, D.: An exploratory analysis of the value of the skills of it personnel: their relationship to is infrastructure and competitive advantage. Decis. Sci. **32**(1), 21–54 (2001)
39. Cao, L., Ramesh, B., Abdel-Hamid, T.: Modeling dynamics in agile software development. Trans. Manag. Inf. Syst. **1**(1), 5 (2010)
40. Capretz, L.: Are software engineers really engineers? World Trans. Eng. Technol. Educ. **1**(2), 233–235 (2002)
41. Capretz, L.: Personality types in software engineering. Int. J. Hum. Comput. Stud. **58**(2), 207–214 (2003)
42. Capretz, L., Ahmed, F.: Making sense of software development and personality types. IT Prof. **12**(1), 6–13 (2010)
43. CAPT: Estimated frequencies of the types in the United States population (2010)
44. Choi, K., Fadi, D., Im, I.: Exploring the underlying aspects of pair programming: the impact of personality. Inf. Softw. Technol. **50**, 1114–1126 (2008)

45. Chow, T., Cao, D.: A survey study of critical success factors in agile software projects. J. Syst. Softw. **81**(6), 961–971 (2008)
46. Christensen, C.: The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail. Harvard Business Review Press, Boston (2013)
47. Christopher, M.: The agile supply chain: competing in volatile markets. Ind. Mark. Manag. **29**(1), 37–44 (2000)
48. Cockburn, A.: Disciplined learning: the successor to risk management. http://www.crosstalkonline.org/storage/issue-archives/2014/201407/201407-Cockburn.pdf (2014). Accessed 1 Oct 2014
49. Cockburn, A.: Characterizing people as non-linear, first order components in software development. In: International Conference on Software Engineering. (1999)
50. Cockburn, A.: Agile Software Development: The Cooperative Game. Addison-Wesley, Boston (2006)
51. Cockburn, A., Highsmith, J.: Agile software development 2: the people factor. IEEE Comput. **34**(11), 131–133 (2001)
52. Cohn, M.: Agile Estimation and Planning. Prentice Hall, New Jersey (2010)
53. Connelly, K.: The growing role of the board in risk oversight. https://www.spencerstuart.com/research-and-insight/the-growing-role-of-the-board-in-risk-oversight (2010). Accessed 1 Oct 2014
54. Coplien, J., Harrison, N.: Organizational Patterns of Agile Software Development. Pearnson Prentice Hall, London (2005)
55. COSO: COSO enterprise risk management—integrated framework. Committee of Sponsoring Organizations of the Threadway Commission (2004)
56. Craddock, A.: Agile Project Management and Scrum (Pocketbook). DSDM Consortium (2013)
57. Crosby, P.: Quality Without Tears. McGraw Hill, New York (1984)
58. Cusumano, M., Yoffie, D.: Software development on internet time. Computer **32**(10), 60–69 (1999)
59. Dallas, M., Clackworthy, S.: Management of Value. The Stationery Office, London (2010)
60. D'aveni, R.: Hypercompetition. Simon and Schuster, New York (2010)
61. Davenport, T., Prusak, L.: Working Knowledge: How Organizations Manage What They Know. Harvard Business Press, Boston (1998)
62. Davis, S.: Managing Corporate Culture. Ballinger Publishing Company, Cambridge (1984)
63. Dayananda, D., Irons, R., Harrison, S., Herbohn, J., Rowland, P.: Capital Budgeting: Financial Appraisal of Investment Projects. Cambridge University Press, Cambridge (2002)
64. De Charms, R.: Personal Causation: The Internal Affective Determinants of Behavior. Routledge, New York (2013)
65. Deming, W.: The New Economics: For Industry, Government, Education. MIT Press, Cambridge (2000)
66. Derby, E., Larsen, D.: Agile Retrospectives: Making Good Teams Great! Pragmatic Bookshelf, Raleigh (2006)
67. DiMaggio, P., Powell, W.: The iron cage revisited: institutional isomorphism and collective rationality in organizational fields. Am. Sociol. Rev. **48**, 147–160 (1983)
68. Dirks, K.: The effects of interpersonal trust on work group performance. J. Appl. Psychol. **84**(3), 445–455 (1999)
69. Driskell, J., Hogan, R., Salas, E.: Personality and Group Performance. Sage Publications, Beverly Hills (1987)
70. DSDM Consortium: The Agile PMO (Pocketbook). DSDM Consortium (2012)
71. DSDM Consortium: Agile project management handbook. DSDM Consortium (2014)
72. DSDM Consortium: The DSDM agile programme management handbook. DSDM Consortium (2014)
73. DSDM Consortium: The DSDM agile project framework handbook. DSDM Consortium (2014)

74. Duvall, P., Matyas, S., Glover, A.: Continuous Integration: Improving Software Quality and Reducing Risk. Pearson Education, Boston (2007)
75. Eckstein, J.: Agile Software Development in the Large: Diving Into the Deep. Pearson Education, Boston (2004)
76. Eisenhardt, K.: Making fast strategic decisions in high-velocity envirnoments. Acad. Manag. J. **32**(3), 543–576 (1989)
77. Eisenhardt, K., Martin, J.: Dynamic capabilities: what are they? Strateg. Manag. J. **21**, 1105–1121 (2000)
78. Eisenhardt, K., Sull, D.: What is strategy in the new economy? Harv. Bus. Rev. **79**(1), 106–117 (2001)
79. Eysenck, H.: Creativity and personality: suggestions for a theory. Psychol. Inq. **4**(3), 147–178 (1993)
80. Feist, G.: A meta-analysis of personality in scientific and artistic creativity. Pers. Soc. Psychol. Rev. **2**(4), 290–309 (1998)
81. Fenton-O'Creevy, M.: Employee involvement and the middle manager: evidence from a survey of organizations. J. Organ. Behav. **19**(1), 67–84 (1998)
82. Fernald, L.: A new trend: creative and innovative corporate environments. J. Creative Behav. **23**(3), 208–213 (1989)
83. Fowler, M.: Refactoring: Improving the Design of Existing Code. Addison-Wesley, Boston (1997)
84. Fowler, M.: FeatureBranch. http://martinfowler.com/bliki/FeatureBranch.html (2009). Accessed 1 Oct 2014
85. French, R., Rayner, C., Rees, G., Rumbles, S.: Organizational Behaviour. Wiley, New York (2011)
86. Garland, R.: Project Governance: A Practical Guide to Effective Project Decision Making. Kogan Page Publishers, Brisbane (2009)
87. Garland, D., Fairbanks, G.: Just Enough Architecture: A Risk-Driven Approach. Marshall & Brainerd, Boulder (2010)
88. Garvin, D.: What does 'Product Quality' really mean? (1984). Accessed 22 June 2014
89. Garvin, D.: Competing on the eight dimensions of quality. Harv. Bus. Rev. **65**, 101–109 (1987)
90. Georgsdottir, A., Getz, I.: How flexibility facilitates innovation and ways to manage it in organizations. Creativity Innov. Manag. **13**(3), 166–175 (2004)
91. Gharajedaghi, J.: Systems Thinking: Managing Chaos and Complexity: A Platform for Designing Business Architecture. Elsevier, Amsterdam (2011)
92. Gilmore, J., Pine, B.: The four faces of mass customization. Harv. Bus. Rev. **75**(1), 91–101 (1996)
93. Golden, W., Powell, P.: Towards a definition of flexibility: in search of the Holy Grail? Omega **28**(4), 373–384 (2000)
94. Goleman, D.: Emotional Intelligence: Why It Can Matter More than IQ. Bloomsburry, London (1995)
95. Granovetter, M.: Economic action and social structure: the problem of embeddedness. Am. J. Sociol. **91**, 481–510 (1985)
96. Grant, R.: The resource-based view of competitive advantage: implications for strategy formulation. Calif. Manag. Rev. **33**, 114–135 (1991)
97. Günsel, A., Açijgöz, A.: The effects of team flexibility and emotional intelligence on software development performance. Group Decis. Negot. **22**, 359–377 (2013)
98. Hackman, J., Oldham, G.: Motivation through the design of work: test of a theory. Organ. Behav. Hum. Perform. **16**(2), 250–279 (1976)
99. Hackman, J., Morris, C.: Group tasks, group interaction process and group performance outcomes: a review and proposed integration. Adv. Exp. Soc. Psychol. **8**, 45–99 (1975)
100. Hamel, G.: The why, what, and how of management innovation. Harv. Bus. Rev. **84**(2), 72 (2006)
101. Harris, E.: Strategic Project Risk Appraisal and Management. Gower Publishing Ltd., London (2012)

102. Harvey, R.: Reliability and Validity, in MBTI Applications, pp. 5–29. Consulting Psychologists, Palo Alto (1996)
103. Hazzan, O., Dublinsky, Y.: Agile Software Engineering. Springer, London (2008)
104. Heron, J., Page, K.: The Complete Facilitators Handbook. Kogan Page, London (1999)
105. Herzberg, F.: The motivation-hygiene concept and problems of manpower. Pers. Adm. **27**, 3–7 (1964)
106. Highsmith, J.: Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Addison-Wesley, Reading (2013)
107. Highsmith, J.: Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Addison-Wesley, Reading (2013)
108. Hikaka, M.: Is extreme programming just old wine in New Bottles: a comparison of two cases. http://www.sirel.fi/ttt/Downloads/Old%20Wine%20in%20A%20New%20Bottle.pdf (2005). Accessed 1 Oct 2014
109. Hillson, D.: Managing Risk in Projects. Gower, Farnham (2009)
110. HM Treasury: The Orange Book: Management of Risk—Principles and Concepts. Stationery Office Books, London (2004)
111. Hoda, R., Noble, J., Marshall, S.: Organizing self-organizing teams. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, vol. 1, pp. 285–294 (2010)
112. Hoda, R., Noble, J., Marshall, S.: Supporting self-organizing agile teams—what's senior management got to do with it? Proceedings International Conference. Agile Processes in Software Engineering and Extreme Programming, pp. 73–87 (2011)
113. Hoda, R., Noble, J., Marshall, S.: Self-organizing roles on agile software development teams. IEEE Trans. Softw. Eng. **39**(3), 422–444 (2013)
114. Hoda, R., Noble, J., Marshall, S.: Impact of inadequate customer involvement on self-organizing agile teams. J. Inf. Softw. Technol. **53**(2), 521–534 (2011)
115. Hoegl, M., Parboteeah, P.: Autonomy and teamwork in innovative projects. Hum. Resour. Manag. **45**(1), 67–79 (2006)
116. Hoegl, M., Parboteeah, K.: Creativity-relevant skills and the task performance of innovation teams: how teamwork maters. J. Eng. Technol. Manage. **24**, 148–166 (2007)
117. Hofstede, G.: Culture's Consequences: Comparing Values, Behaviors, Institutions and Organizations Across Nations. Sage, Thousand Oaks (2003)
118. Hofstede, G.: Dimensions—geert hofstede. http://geert-hofstede.com/dimensions.html (2013). Accessed 1 Oct 2014
119. Holland, J.: Complex adaptive systems. Daedalus **121**, 17–30 (1992)
120. Holling, C.: Myths of ecological stability. Studies in Crisis Management. Butterworth, Montreal (1979)
121. Hope, J., Fraiser, R.: Beyond Budgeting: How Managers Can Break Free from the Annual Performance Trap. Harvard Business School Press, Boston (2003)
122. Hunt, A., Thomas, D.: The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley Professional, Reading (2000)
123. Iansiti, M.: Shooting the rapids: manaing product development in turbulent environments. Calif. Manag. Rev. **38**(1), 37–58 (1995)
124. IBM: Disciplined Agile Delivery: An Introduction. https://www.ibm.com/developerworks/community/blogs/ambler/entry/disciplined_agile_delivery_an_introduction_white_paper22?lang=en (2011). Accessed 1 Oct 2014
125. IBM: Form 10-K, Annual report: international business machines corporation.https://www.sec.gov/Archives/edgar/data/51143/000104746914001302/a2217495z10-k.htm (2014)
126. IFRS: About the IFRS foundation and the IASB. http://www.ifrs.org/About-us/Pages/IFRS-Foundation-and-IASB.aspx (2014)
127. Imai, M.: Gemba Kaizen: A Commonsense, Low-Cost Approach To Management. McGraw-Hill, New York (1997)
128. Imai, K., Ikujiro, N., Takeuchi, H.: Managing the new product development process: how Japanese companies learn to unlearn. The Uneasy Alliance: Managing the Productivity-Technology Dilemma, pp. 377–375. (1985)

129. ISACA: The risk IT framework. ISACA (2009)
130. ISACA: The rsk IT practitioner guide. ISACA (2009)
131. ISACA: COBIT 5: A business framework for the governance and management of enterprise IT. ISACA (2012)
132. ISO: ISO 31000:2009 Risk management—Principles and guidelines. International Standards Organization (2009)
133. ISO, B.: 9000: 2005 Quality management systems. Fundamentals and vocabulary, British Standards Institution (2005)
134. Jeffries, R.: XProgramming. http://xprogramming.com/index.php (2013). Accessed 1 Oct 2014
135. Johnson, G.: Strategic Change and the Management Process. Blackwell, Oxford (1987)
136. Jones, C.: Measuring defect potentials and defect removal efficiency. CrossTalk J. Defense Softw. Eng. **21**(6), 11–13 (2008)
137. Jones, T.: Estimating Software Costs. McGraw-Hill Inc., New York (1998)
138. Jordan, P., Ashkanasy, N., Hartel, C.: Emotional intelligence as a moderator of emotional and behavioural reactions to job insecurity. Acad. Manag. Rev. **27**, 361–372 (2002)
139. Jordan, P., Ashlea, C.: Managing emotions during team problem solving: emotional intelligence and conflict resolution. Hum. Perform. **17**(2), 195–218 (2004)
140. Juran, J., Gryna, F.: Juran's Quality Control Handbook. McGraw Hill, New York (1988)
141. Kano, N., Seraku, N., Takahashi, F., Tsuji, S.: Attractive quality and must-be quality. J. Jpn. Soc. Qual. Control **41**, 39–48 (1984)
142. Karn, J., Syed-Abdullah, S., Cowling, A., Holcombe, M.: A study into the effects of personality type and methodology on cohesion in software engineering teams. Behav. Inf. Technol. **26**(2), 99–111 (2007)
143. Kelly, J., Barsade, S.: Mood and emotions in small groups and work teams. Organ. Behav. Hum. Decis. Process. **86**(1), 99–130 (2001)
144. Kichuk, S., Wiesner, W.: The big five personality factors and team performance: implications for selecting successful product design teams. J. Eng. Technol. Manage. **14**(3–4), 195–221 (1997)
145. Knight, F.: Risk. Uncertainty and Profit. Houghton Mifflin, Boston (1921)
146. Kotter, J.: Leading Change. Harvard Business Press, Boston (1996)
147. Langfred, C.: The paradox of self-management: Individual and group autonomy in work groups. J. Organ. Behav. **21**(5), 563–585 (2000)
148. Le Clair, C.: Make business agility a key corporate attribute—it could be what saves you. http://blogs.forrester.com/craig_le_clair/13-09-09-make_business_agility_a_key_corporate_attribute_it_could_be_what_saves_you. Accessed 1 Oct 2014
149. Lea, G., Xia, W.: Relationships between software team flexibility, autonomy, diversity and project performance. Philadelphia OCIS (2007)
150. Lee, G., Xia, W.: The ability of information systems development project teams to respond to business and technology changes: a study of flexibility measures. Eur. J. Inf. Syst. **14**(1), 75–92 (2005)
151. Leffingwell, D.: Scaling Software Agility: Best Practices for Large Enterprises. Pearson Education, Boston (2007)
152. Li, Y., Chang, K., Chen, H., Jiang, J.: Software development teams flexibility antecdents. J. Syst. Softw. **83**, 1726–1734 (2010)
153. Lines, M., Ambler, S.: Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. IBM Press, Upper Saddle River (2012)
154. Loeliger, J., McCullough, M.: Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development. O'Reilly Media, Inc., USA (2012)
155. Luna, A., Costa, C., Moura, H., Novaes, M., do Nascimento, C.: Agile governance in information and communication technologies: shifting paradigms. J. Inf. Syst. Technol. Manag. **7**(2), 311–334 (2010)
156. Lyons, M.: The DP psyche. Datamation **31**(16), 103–105 (1985)

157. MacComack, A., Verganti, R., Iansiti, M.: Developing products on internet time: the anatomy of a flexible development process. Manag. Sci. **47**(1), 133–150 (2001)
158. Mangalaraj, G. and Bhadauria, V.: Pair programming: effects of trust on software quality. AMCIS p. 117ff. (2007)
159. Manz, C.: Self-leading work teams: moving beyond self-management myths. Hum. Relat. **45**(11), 1119–1140 (1992)
160. March, J.: Exploration and exploitation in organzational learning. Organ. Sci. **2**(1), 71–87 (1991)
161. Martin, J.: Rapid Application Development. Macmillan, USA (1991)
162. Maruping, L., Zhang, X., Venkatesh, V.: Role of collective ownership and coding standards in coordinating expertise in software project teams. Eur. J. Inf. Sci. **18**, 335–371 (2009)
163. Maslow, A., Frager, R., Fadiman, J., McReynolds, C.: Motivation and Personality. Harper & Row, New York (1970)
164. Mason-Jones, R., Naylor, B., Towill, D.: Engineering the leagile supply chain. Int. J. Agile Manag. Syst. **2**(1), 54–61 (2000)
165. Mazni, O., Syed-Abdullah, S., Hussin, N.: Analyzing personality types to predict team performance. In: International Conference on Science and Social Research, pp. 624–628. (2010)
166. McCall, J., Richards, P., Walters, G.: Factors in Software Quality, vol. i-iii. Tech. rep., DTIC Document (1977)
167. McClelland, D.: The Achieving Society. Simon & Schuster, New York (1967)
168. McCrae, R., Costa, P.: Reinterpreting the Myers-Briggs type indicator from the perspective of the five-factor model of personality. J. Pers. **57**(1), 17–40 (1989)
169. McCrae, R., Costa, P.: Personality in Adulthood: A Five-Factor Theory Perspective. Guilford Press, New York (2003)
170. McCrae, R., Costa, P., Busch, C.: Evaluating comprehensiveness in personality systems: The Californian Q-set and the five-factor model. J. Pers. **54**, 430–446 (1986)
171. McCrae, R., John, O.: An introduction to the five-factor model and its applications. J. Pers. **60**(2), 175–215 (1992)
172. Meyer, B.: Object-Oriented Software Construction, vol. 2. Prentice Hall, Englewood Cliffs (1988)
173. Meyer, J., Rowan, B.: Institutionalized organizations: formal structure as myth and ceremony. Am. J. Sociol. **83**, 340–363 (1977)
174. Meyer, J., Allen, N.: A three-component conceptualization of organizational commitment. Hum. Resour. Manag. Rev. **1**(1), 61–89 (1991)
175. Meyerson, D., Martin, J.: Cultural change: an integration of three different views. J. Manag. Stud. **24**(6), 623–647 (1987)
176. Miller, G.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol. Rev. **63**(2), 81–97 (1956)
177. Mills, H.: Software Productivity. Dorset House, New York (1988)
178. Mintzberg, H.: The Strategy Concept 1: Five p's for Strategy. University of California, California (1987)
179. Mintzberg, H., Waters, J.: Of strategies, deliberate and emergent. Strateg. Manag. J. **6**(3), 257–272 (1985)
180. Moe, N., Dingsøyr, T., Dybå, T.: Understanding self-organizing teams in agile software development. In: IEEE 19th Australian Conference on Software Engineering, pp. 76–85. (2008)
181. Moeller, R.: COSO Enterprise Risk Management: Establishing Effective Governance, Risk and Compliance Processes. Wiley, New York (2011)
182. Mohammed, S. and Angell, L.: Personality heterogeneity in teams: which differences make a difference for team performance? Small Group Res. **34**(6), 651–677 (2003)
183. Moran, A.: Agile Charting: A Practical Guide. Institute for Agile Risk Management (2014)
184. Moran, A.: Agile Risk Management. Springer, Berlin (2014)
185. Moran, A.: Agile Risk Management and DSDM (Pocketbook). DSDM Consortium (2015)
186. Morgan, G.: Images of Organization. Sage Publications, Thousand Oaks (1986)

187. Mowday, R., Lyman, P., Steers, R.: Employee-Organization Linkages: The Psychology of Commitment, Absenteeism, and Turnover. Academic Press, New York (1982)
188. Müller, R.: Project Governance. Gower, Aldershot (2012)
189. Murphy, T., Duggan, J., Norton, D., Prentice, B., Plummer, D.C., Landry, S.: Predicts 2010: Agile and Cloud Impact Application Development Directions. Gartner (2009)
190. Murray-Webster, R., Hillson, D.: Managing Group Risk Attitude. Gower, Aldershot (2008)
191. Myers-Briggs, I., McCaulley, M., Quenk, N., Hammer, A.: MBTI Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator® Instrument. Consulting Psychologists Press, Palo Alto (2009)
192. Naylor, B., Naim, M., Berry, D.: Leagility: integrating the lean and agile manufacturing paradigms in the total supply chain. Int. J. Prod. Econ. **62**(1), 107–118 (1999)
193. Nerur, S., Balijepally, V.: Theoretical reflections on agile development methodologies. Commun. ACM **50**(3), 79–83 (2007)
194. Nyfjord, J.: Towards integrating agile development and risk management. Ph.D. thesis, Stockholm (2008)
195. Office of Government Commerce: Management of risk: Guide for practitioners. The Stationery Office (2007)
196. OGC: Managing successful projects with PRINCE2. Stationery Office Books (2009)
197. OGC: ITIL V3 foundation handbook. Stationery Office Books (2011)
198. OGC: Managing successful programmes. Stationery Office Books (2011)
199. O'Sullivan, B.: Mercurial: The Definitive Guide. O'Reilly Media, Inc., Sebastopol (2009)
200. Palmer, S., Felsing, J.: A practical guide to feature-driven development. Prentice Hall, Upper Saddle River (2002)
201. Patton, J.: Don't know what I want, but I know how to get it. http://www.agileproductdesign.com/blog/dont_know_what_i_want.html (2008). Accessed 1 Dec 2013
202. Pettigrew, A.: The Politics of Organizational Decision-making. Routledge, London (2014)
203. Pichler, R.: Agile Product Management: Creating Products That Customers Love. Addison-Wesley, Amsterdam (2010)
204. Pink, D.: Drive: The Surprising Truth About What Motivates Us. Penguin, New York (2011)
205. Pittenger, D.: Measuring the MBTI... and coming up short. J. Career Plan. Employ. **54**(1), 48–52 (1993)
206. Poppendieck, M., Poppendieck, T.: Lean Software Development: An Agile Tookit. Addison-Wesley, New York (2003)
207. Porter, M.: Competitive Strategy: Techniques for Analyzing Industries and Competitors. Free Press, New York (1980)
208. Pries-Heje, M.: Racing the e-bomb: How the internet is redefining information systems development methodology. In: Realigning Research and Practice in Information Systems Development, pp. 49–68. Springer, New York (2001)
209. Project Management Institute: Practice standard for project configuration management. Project Management Institute (2007)
210. Project Management Institute: Practice standard for project risk management. Project Management Institute, Incorporated (2009)
211. Project Management Institute: A Guide to the Project Management Body of Knowledge: PMBOK Guide. Project Management Institute, Incorporated (2013)
212. Project Management Institute: A Guide to the Project Management Body of Knowledge: PMBOK Guide. Project Management Institute, Incorporated (2014)
213. Protiviti: Guide to enterprise risk management. http://www.protiviti.com/en-US/Pages/Guide-to-Enterprise-Risk-Management.aspx (2013). Accessed 1 Oct 2014
214. Remenyi, D., Money, A., Bannister, F.: The Effective Measurement and Management of ICT Costs and Benefits. Elsevier, Burlington (2007)
215. Reus, T., Liu, Y.: Rhyme and reason: emotional capability and the performance of knowledge-intensive work groups. Hum. Perform. **17**(2), 245–266 (2004)
216. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Random House, New York (2011)

217. Robinson, H., Sharp, H.: Organisational culture and xp: three case studies. Proc. Agile Conf. **1**, 49–58 (2005)
218. Runco, M., Pritzker, S.: Encyclopedia of Creativity, vol. 2. Elsevier, Burlington (1999)
219. Ruuska, I., Artto, K., Aaltonen, K., Lehtonen, P.: Dimensions of distance in a project network: exploring olkiluoto 3 nuclear power plant project. Int. J. Proj. Manag. **27**(2), 142–153 (2009)
220. Ryan, R., Deci, E.: Intrinsic and extrinsic motivations: classic definitions and new directions. Contemp. Educ. Psychol. **25**(1), 54–67 (2000)
221. Ryan, R., Deci, E.: Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Am. Psychol. **55**(1), 68 (2000)
222. Ryff, C., Keyes, C.: The structure of psychological well-being revisited. J. Pers. Soc. Psychol. **69**(4), 719–727 (1995)
223. Sach, R. and Petre, M. and Sharp, H.: The use of MBTI in software engineering. In: 22 Annual Psychology of Programming Interest Group. (2010)
224. SAFe: Scalable agile framework. http://scaledagileframework.com/ (2013). Accessed 1 Oct 2014
225. SAFe: Capitalisation of software development costs in a SAFe environment. http://scaledagileframework.com/capitalization/ (2014)
226. Salovey, P., Mayer, J.: Emotional intelligence. Imagination Cogn. Pers. **9**(3), 185–211 (1989)
227. Say, M.: Managers must become IT-savvy. http://www.ft.com/intl/cms/s/2/ae1b6c0a-6362-11e4-8a63-00144feabdc0.html#axzz3J3cxOeXmdd
228. Schein, E.: Organizational Culture and Leadership, vol. 2. Wiley, New York (2010)
229. Schermerhorn, C.: Management, Wiley (2002)
230. Schmit, M., Ryan, A., Stierwalt, S., Powell, A.: Frame-of-reference effects on personality scale scores and criterion-related validity. J. Appl. Psychol. **80**(5), 607–620 (1995)
231. Schumpeter, J.: Capitalism, Socialism and Democracy. Routledge, London (2013)
232. Schwaber, K., Kong, P., Starr, D.: Evidence-based management guide: empirical management for software organizations. http://www.ebmgt.org/portals/agilitypath/Documents/EBMgt_Guide_v1_CWT.pdf
233. Schwaber, K., Sutherland, J.: The Scrum Guide. Scrum.orgr, Boston (2013)
234. Schwaber, K.: Agile Project Management with Scrum. Microsoft Press, Redmond (2004)
235. Schwaber, K.: Scrum is hard and disruptive. http://www.controlchaos.com/storage/scrum-articles/Scrum%20Is%20Hard%20and%20Disruptive.pdf (2006). Accessed 1 Oct 2014
236. Scott, W.: Organizations: Natural, Rational and Open Systems. Prentice-Hall, Upper Saddle River (1998)
237. Scott, A.: The agile unified process (AUP). http://www.ambysoft.com/unifiedprocess/agileUP.html (2005). Accessed 1 Oct 2014
238. Séguin, N., Abran, A., Dupuis, R.: Software engineering principles: a survey and an analysis. In: Proceedings of the ACM Third C* Conference on Computer Science and Software Engineering, pp. 59–65. (2010)
239. Séguin N., Tremblay, G., Bagane, H.: Agile principles as software engineering principles: an analysis. In: Agile Processes in Software Engineering and Extreme Programming, pp. 1–15. Springer, Berlin (2012)
240. Senge, P.: The Art and Practice of the Learning Organization. Doubleday, New York (2000)
241. Sersik, S., Schröder, G.: Automated integration testing in agile environments. Agile Testing, p. 61. http://www.gschroed.de/Downloads/testingexperience03_09_Sersik_Schroeder.pdf (2009)
242. Siakas, K., Siakas, E.: The agile professional culture: a source of agile quality. Softw. Process Improv. Pract. **12**(6), 597–610 (2007)
243. Sircus, M.: Nano medicine treatments for antibiotic resistant bacteria. http://drsircus.com/medicine/nano-medicine-treatments-antibiotic-resistant-bacteria (2014). Accessed 1 Oct 2014
244. Sitkin, S., Pablo, A.: Reconceptualizing the determinants of risk behavior. Acad. Manag. Rev. **17**, 9–38 (1992)
245. Sminia, H.: The Strategic Manager. Routledge, London (2014)

246. Smith, D.: The personality of the systems analyst: an investigation. ACM SIGCPR Comput. Pers. **12**(2), 12–14 (1989)
247. Smith, D., Hanges, P., Dickson, M.: Personnel selection and the five-factor model: reexamining the effects of applicant's frame of reference. J. Appl. Psychol. **86**(2), 304–315 (2001)
248. Sommerville, I.: Software Engineering. Wiley, New York (2004)
249. Stephen, J.: System error: fixing the flaws in government IT. http://www.instituteforgovernment.org.uk/sites/default/files/publications/System%20Error.pdf (2011)
250. Stevenson, H., Gumpert, D.: The heart of entrepreneurship. Harv. Bus. Rev. **63**(2), 85–94 (1985)
251. Stricker, L., Ross, J.: An assessment of some structural properties of the Jungian personality typology. J. Abnorm. Soc. Psychol. **68**, 62–71 (1964)
252. Suchman, M.: Managing legitimacy: strategic and institutional approaches. Acad. Manag. Rev. **20**(3), 571–610 (1995)
253. Sutherland, J.: Iterative vs. incremental development. http://scrum.jeffsutherland.com/2010/01/iterative-vs-incremental-development.html (2010). Accessed 1 Oct 2014
254. Takeuchi, H., Nonaka, I.: The new new product development game. Harv. Bus. Rev. **64**(1), 137–146 (1986)
255. Takeuchi, H., Nonaka, I.: New new product development game. Har. Bus. Rev. **96116**, 137–146 (1986)
256. Takeuchi, H., Nonaka, I.: The Knowledge Creating Company. Oxford University Press, Oxford (2013)
257. The Open Group: Welcome to TOGAF® version 9.1 enterprise edition (2013). Accessed 22 June 2014
258. Thompson, M.: Cultural Theory. Westview, Boulder (1990)
259. Tolfo, C., Wazlawick, R., Ferreira, M., Forcellini, F.: Agile methods and organizational culture: reflections about cultural levels. J. Softw. Maintenance Evol. Res. Pract. **23**(6), 423–441 (2011)
260. Tudor, D.: Agile Project and Service Management: Delivering IT Services Using ITIL, PRINCE2 and DSDM Atern. Stationery Office Books, London (2010)
261. Turner, J., Simister, S.: Project contract management and a theory of organization. Int. J. Proj. Manag. **19**(8), 457–464 (2001)
262. Verganti, R.: Leveraging on systemic learning to manage early phases of product innovation projects. R D Manag. **27**(4), 377–392 (1997)
263. Verganti, R.: Planned flexibility: linking anticipation and reaction in production development projects. J. Prod. Innov. Manag. **16**(4), 363–376 (1999)
264. Vroom, V.: Work and Motivation. Wiley, New York (1964)
265. Wake, B.: INVEST in good stories, and SMART tasks. http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/ (2003)
266. Weinberg, G.: Quality Software Management: Systems Thinking, vol. 1. Dorset House Publishing, New York (1992)
267. Weinberg, G.: Quality Software Management: Systems Thinking, vol. 1. Dorset House Publishing Co., Inc., New York (1992)
268. Weinberg, G.: An Introduction to General Systems Thinking. Dorset House, New York (2001)
269. Wells, D.: The rules of extreme programming. http://www.extremeprogramming.org/rules.html (1999). Accessed 1 Oct 2014
270. Wells, D.: Extreme programming project. http://www.extremeprogramming.org/map/project.html (2000). Accessed 1 Oct 2014
271. Wells, D.: Extreme programming: a gentle Introduction. http://www.extremeprogramming.org (2009). Accessed 1 Oct 2014
272. Wells, D.: Iterative planning. http://www.agile-process.org/iterative.html (2009). Accessed 1 Oct 2014
273. Wemerfelt, B.: A resource-based view of the firm. Strateg. Manag. J. **5**, 171–180 (1984)
274. Wendorff, P.: An essential distinction of agile software development processes based on systems thinking in software engineering management. In: Third International Conference on eXtreme Programming and Agile Processes in Software Engineering. (2002)

275. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press, Cambridge (1998)
276. Westbrook, P.: Frequencies of MBTI types among computer technicians. J. Psychol. Type **15**, 50–53 (1988)
277. White, K.: A preliminary investigation of information systems team structures. Inf. Manag. **7**(6), 331–335 (1984)
278. Williams, O.: Markets and Hierarchies: Analysis and Antitrust Implications. Macmillan, London (1975)
279. Winch, G.: The governance of project coalitions: towards a research agenda. Commercial Management of Projects: Defining the Discipline, pp. 324–43. Blackwell Publishing, Oxford (2006)
280. Wolff, S., Anthony, T., Vanessa, U.: Emotional intelligence as the basis of leadership emergence in self-managing teams. Leadersh. Q. **13**(5), 222–505 (2002)
281. Womack, J., Jones, D.: The Machine that Changed the World. Simon & Schuster, New York (1996)
282. Womack, J., Jones, D., Roos, D.: The Machine that Changed the World. Simon & Schuster, New York (2008)
283. Xu, X.: Position of customer order decoupling point in mass customization. Int. Conf. Mach. Learn. Cybern. **1**, 302–307 (2007)
284. Yang, K., Bond, M.: Exploring implicit personality theories with indigenous or imported constructs: the Chinese case. J. Pers. Soc. Psychol. **58**, 1087–1095 (1990)

Index

A. Moran, *Managing Agile*, DOI 10.1007/978-3-319-16262-1