Addressing 'agility' in current project management standards and other authoritative publications [1]

By Robert Buttrick

The purpose of this paper

The purpose of this paper is to examine how the term 'agile' and its derivatives are used in a range of standards and other authoritative project management related publications with a view to understanding and reconciling the apparent differences. The following documents are the primary sources used:

- ANSI/PMI 99-001-2021, The Standard for Project Management, 2021
- APM Body of Knowledge, 7th edition, 2019
- PRINCE2®, Projects in a controlled environment, 6th edition, 2017
- PRINCE2® Agile, 1st edition, 2015
- BS 6079:2019. Project management – Principles and guidance for the management of projects (2019)
- GovS 002, 2019, Project delivery functional standard, v2, 2021
- IPMA standards –Baselines (PEB, ICB, OCB, ICB4)
- ISO 21500 series of standards (ISO 21502:2020, ISO 21503:2022, ISO 21504:2022, 21505:2017)
- PM^2, Project Management Methodology, Guide 3.0, 2018
- PM^2 Agile, v3.01, 2021
- PMBOK® Guide, A Guide to the Project Management Body of knowledge, 7th edition, 2021

Appendices C to K to this paper include a commentary on these publications to give the reader specific references to the respective texts. Other documentation is referred to in this paper to provide context and alternative viewpoints.

Different perspectives of 'agile'

The term 'agile' is used in various contexts and whilst being initially and commonly associated with software development, its use has expanded to encompass entire organisations, sometimes referred to as 'business agility'. It seems everyone wants their business or work to be 'agile', which is hardly surprising as some antonyms for the word include 'clumsy', 'dull' and 'boring'!

The uses of the term 'agile' generally fall into three categories[2]:

[1] How to cite this paper: Buttrick, R. (2022). Addressing 'agility' in current project management standards and other authoritative publications, *PM World Journal*, Vol. XI, Issue XII, December.

[2] Source PRINCE2® Agile (adapted)

- **management frameworks**, defining 'how' to do the work. They typically include methods and processes, guidance and codes of practice.
- **techniques**, defining how specific tasks are undertaken, many of which are not unique to 'agile' but have attracted a separate set of jargon (for example, retrospective vs lessons learned review). Techniques are often a sub-set of a management framework.
- **behaviours and mindsets**, which emphasize the culture required for success. It is often said a person (or organisation) should *be* agile, not *do* agile. A good management framework or technique is useless if undertaken with inappropriate mindset and behaviours.

These categories are not mutually exclusive. For example, some methods are designed to work with particular behaviours or mindsets and incorporate specific techniques. Table 4 in Appendix A, summarises how these categories are reflected in the primary documents cited in this paper.

Over the 20 years since the term 'agile' has been in use, perceptions of 'project management' have moved from one where the aim is to deliver a defined scope, on time and to budget (as in PMI's triple constraint) to one which focuses on outcomes and benefits (as in ISO 21502, BS 6079, GovS 002 and PRINCE2®) which places the assessment of success associated with those who are the object of the organisational or societal changes triggered by a project, rather with those running a project.

Terminology

Words can get in the way of communication

One of the barriers to a common understanding of 'agile' relates to the words used when describing 'agile'. Words are a means of communication and whilst, on first glance, ideas can appear similar across a range of sources, words can be used in distinctive or different ways resulting in diverse meanings. In addition, many 'agile' methods and techniques have their own terminology (some might say 'jargon') and the chances of misunderstanding, misinterpretation and argument increase. Some advocates of 'agile' do not regard such terminology as jargon but as essential, as those words have been specifically chosen to promote and support the mindset and behaviours necessary to using the associated 'agile' processes and techniques.

Often the change of term appears to be a simple substitution of an established term (for example, 'retrospective' for 'lessons learned'). Some terms are registered trademarks and designed to take commercial advantage of the methods and techniques they apply to. Despite many of the practices having been in use for many years (some under different terms), people who do not use the 'new' terms are often considered 'traditional', with the implicit implication their approaches are inadequate and out of date. In some cases, the terms relate to a prescriptive way of doing something (technique, process or method). On the other hand, some see the use of the 'newer terms' as change for change's sake, and an unnecessary repackaging established best practice. (See "Is 'traditional' really 'traditional'", later in the paper).

Looking at the words people use

Because there are so many interpretations of what 'agile' means, alternative descriptions are sometimes used to clarify the author's intent. The terms 'adaptive', 'iterative' and 'incremental' are often used, juxtaposed with 'predictive' and 'linear' approaches. Many common 'agile' terms derive from specific software development methods and focus on software as an output.

Table 5 (in Appendix B) includes dictionary definitions for some terms commonly used in relation to 'agile' approaches.

Table 6 (in Appendix B) provides examples of established project management terms and some 'agile' equivalents. Note, just as there is no definitive glossary of agile terms, there is no definitive publication for such a mapping and Table 6 only provides suggestions, not strict equivalents.

Don't consider the terms used as wrong, but rather 'differently right'

An authoring panel for a standard, body of knowledge, method or technique is free to choose the language and terms to be used. Some organisations, however, place restrictions on this freedom, for very good reasons. For example, authors of International Standards within the same series are required to use the same terms and definitions; this makes sense as doing otherwise would confuse the reader and risk introducing the very ambiguity standards set out to eliminate. The freedom to select terms and create new ones, has however led to a profusion of different meanings, often with little commonality across them. It is not that some terms are wrong, they are simply 'differently right'. It is, however, important that within a publication or series of publications, the chosen terms should be used consistently.

The aim of standards, whether international or national is to promote better business, better regulation and better products and services that consumers can trust. In other words, easing trade and relationships across the world. If people can use and understand the same terms, working together and buying products and services becomes easier, as does the collaborative working needed to develop, deliver and deploy them, that is to say 'projects'.

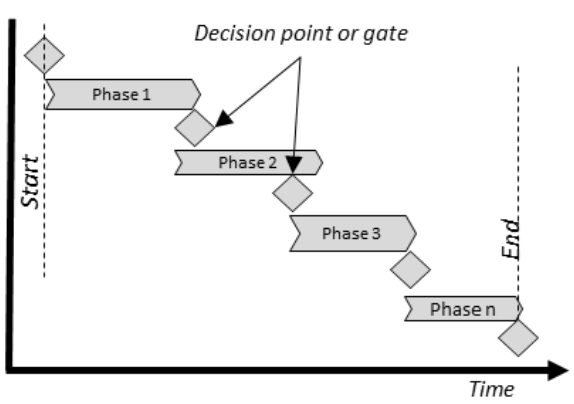
How standards deal with conflicts of terminology

One way for formal standards to deal with a fragmented landscape such as described above, is to use plain language and, wherever possible, adopt the meanings of words described in a widely accepted dictionary, avoiding the use of jargon and trade-marked terms. This serves to bring the ideas and practices, or whatever the topic for the standard is, to a wider readership regardless of context and thereby break down the barriers that using jargon can create. That might not satisfy some people's ambitions to be seen as 'at the cutting edge' but it does, when followed in accordance with drafting rules, lead to the creation of documents which are easily understood, unambiguous, applicable in a wide range of contexts. Importantly, such an approach ensures a document can be easily translated into other languages.

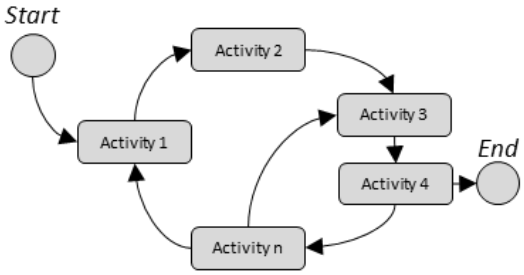
Life cycle or delivery approach?

A key to unlocking the 'agile' deadlock?

In the context of 'agile', one term is particularly problematical: '**life cycle'**. All the primary documents cited in this paper use the term 'life cycle' to be time based where each phase[3] of work happens in a planned time period. Any rework must happen in a current or future phase as it is impossible to travel back in time. On the other hand, there are some publications where 'life cycle' is taken to be iterative, allowing a person to rework something within what they term as a previous 'phase'. In these contexts a life cycle is not time-bound but rather like the usual definitions of a 'process'. In **Figure 1**, the left-hand diagram shows blocks of activities, clustered into phases and when, in time, those phases take place. It provides no information on what is happening in each phase. The right-hand diagram shows a sequence of activities which enable iterations (which can include rework) but provides no information regarding when these activities take place, nor how long they take.



A depiction of a life cycle – governed by time



A depiction of a delivery approach (in this case, a process) – governed by sequence and logic

Figure 1 Depiction of a life cycle and a process

These different perceptions of what 'life cycle' can be a fundamental source of misunderstandings. ISO 21502, BS 6079, GovS 002 and many other authoritative publications, including PRINCE2® and the PMBOK®, make the distinction, shown in **Figure 1** between a **project life cycle** and a **delivery approach**.

- **The project life cycle** is a defined set of phases which may be sequential or overlap and are governed by 'time'. None of the standards cited in this paper, nor PRINCE2®, name the phases, say how many there should be, nor what activities each phase should

[3] PRINCE2® and GovS 002 use the term 'stage' and DSDM uses 'increment'.

include[4], nor what the detailed criteria should be for starting or ending a phase. They simply state that these should be defined and chosen to reflect, among other things, the delivery approach being taken. Each phase can include work which is predictive, iterative, incremental or whatever. Within each phase, justified changes are allowed. If an error is found in the work from a previous phase, a person can't go back in time to make a correction, it is fixed within the current phase or, sometimes, left for a later phase. Each phase builds on whatever the previous phases produced and takes account of new information and requirements (internal or external to the project). A decision can then be taken to redirect the work, if needed. In this sense, this phased approach could be considered adaptive as the direction of the project is progressively steered to respond to new information and the changing context at a macro level at the start of each phase and on a detailed level through change control within each phase (as and when needed). As such, this can be considered 'agile'.

- **delivery approach** is the term used in the primary sources to say what methods, processes, and techniques are used to develop the required deliverables and outputs and embed the required business and/or societal changes, how the team is organised, competencies, behaviours and management culture required and what the appropriate contractual arrangements are (if needed).

The separation of the delivery approach from project life cycle in ISO 21502 is also reflected in the ISO Technical Committee 1's systems and software engineering standards as well as in BS6079, GovS 002, PMBOK® Guide and PRINCE2®, and helps distinguish between 'agile' delivery approaches and project management.

Delivery approach or development approach?

In the primary sources 'delivery approach' (as used in ISO 21502, BS 6079 and GovS 002, PM^2 and PRINCE2®) is referred to in other publications variously as 'development method', 'delivery method' and 'development approach'. The context usually makes the writer's intent clear. The choice of 'delivery approach' as opposed to 'development approach' is because 'development' can have a narrower meaning. For example, 'development' is typically associated with 'developing software' or 'product development', but the outputs and outcomes for projects can be varied, for example frigates and roads are not 'developed'. 'Delivery' was also found to be more accessible for translation, an essential feature for international standards.

Table 1 lists the publications and the words predominantly used as equivalent to ISO 21502's 'delivery approach':

[4] GovS 002 does however include a 'reference life cycle' to illustrate the commonest approaches and enable easy mapping to financial requirements and ensuring options are considered and project's solution is justified. The reference life cycle can be tailored.

Table 1 Use of words to describe the delivery approach

Source	Term used
ISO 21502	delivery approach
BS 6079	delivery approach
GovS 002	delivery approach
ANSI/PMI 99-001-2021	development approach delivery approach
PM^2	delivery approach
IPMA PEB, ICB	agile approach agile development agile development process
APMBoK	development method (used once only) agile method agile approach
PMBOK® Guide	development approach Although all combinations of delivery/development approach/method do appear.
PRINCE2®	delivery approach (Although all combinations of delivery/development approach/method do appear.) Plus agile approach and agile method

Life cycles and how these are explained in various sources

The examples on the following pages highlight how the use of language can either clarify or potentially mislead a reader. The examples underline the need to make the distinction between a project life cycle and a delivery approach in whatever document is being drafted or work being undertaken.

Example 1 from published standards using consistent terminology and concepts

The example here focuses primarily on ISO 21502 and its specific and consistent use of terminology and concepts. ISO 21502 includes the characteristics of a project life cycle in terms of gates (decision points) and phases, which it summarises in a diagram (see **Figure 2**).

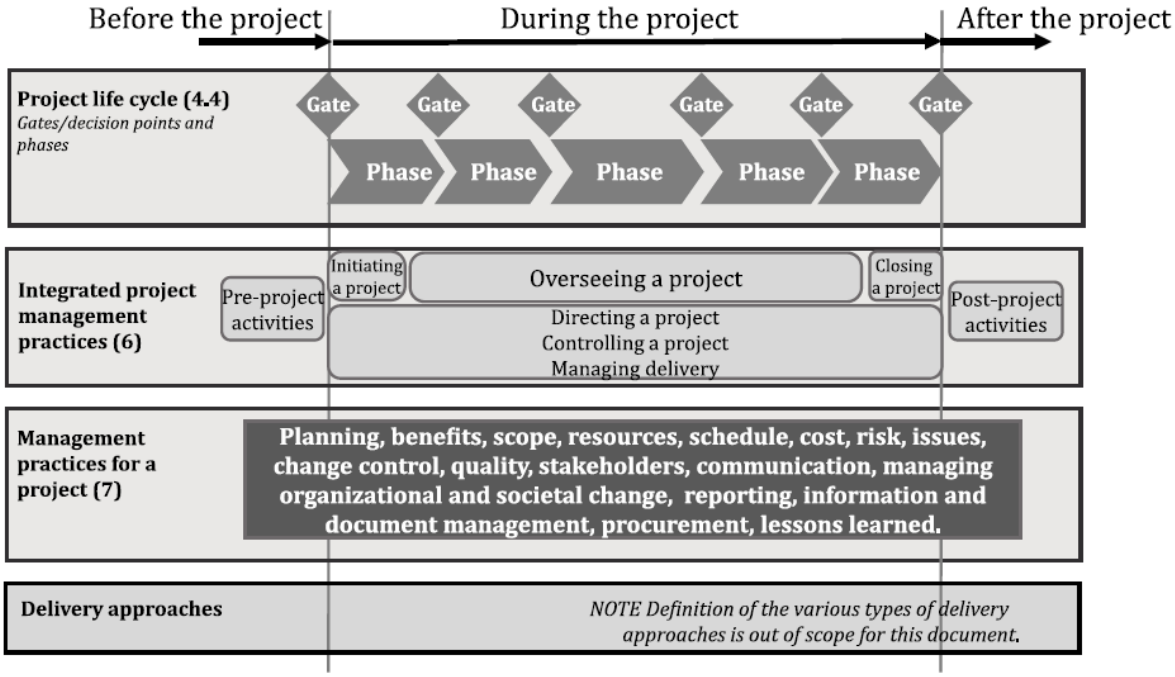


Figure 2 ISO 21502's view of the project life cycle (Source: ISO 21502:2020 Figure 4)

This diagram also shows which management practices are used in each phase and what supporting practices can be drawn on. The 'delivery approach' is explicitly out of scope of this standard. BS 6079, GovS 002 and the PRINCE2® method take a similar approach. None of these sources state what life cycle should be used under what circumstances. In ISO 21502, the 'life cycle' can be designed to reflect the delivery approach being taken, such as predictive, iterative, incremental, adaptive or a hybrid approaches (which it shows at the bottom of the diagram). It allows 'operations' to be within the scope of a project. It elaborates by saying that each phase should have a defined start and end, preceded by a decision point. Often referred to as 'gates', decision points are essential aspects of project governance. Unlike in PRINCE2®, the phases in ISO 21502, BS6079 and GovS 002 need not be strictly sequential but can overlap[5]. As such, the life cycle can encompass any type of project or delivery approach.

Example 2 from academic and textbook depictions

Many academic papers and textbooks attempt to explain the differences between 'agile' approaches versus non-agile approaches (which they sometimes call 'traditional') through a series of diagrams and descriptions, such as shown in **Figure 3**. These tend to progress from a sequential set of 'boxes' (often called 'linear' or 'predictive') to various diagrams with loop backs to different points in the sequence, often termed 'incremental', 'iterative', 'adaptive' and 'extreme' as the loop back retreats further back in the sequence. Confusingly, these diagrams are often headed as 'life cycles', so beware if you see or use them.

[5] AXELOS published a white paper which recommends that the strictly consecutive nature of PRINCE2®'s stages is relaxed. In any case, such a relaxation is permissible under tailoring.

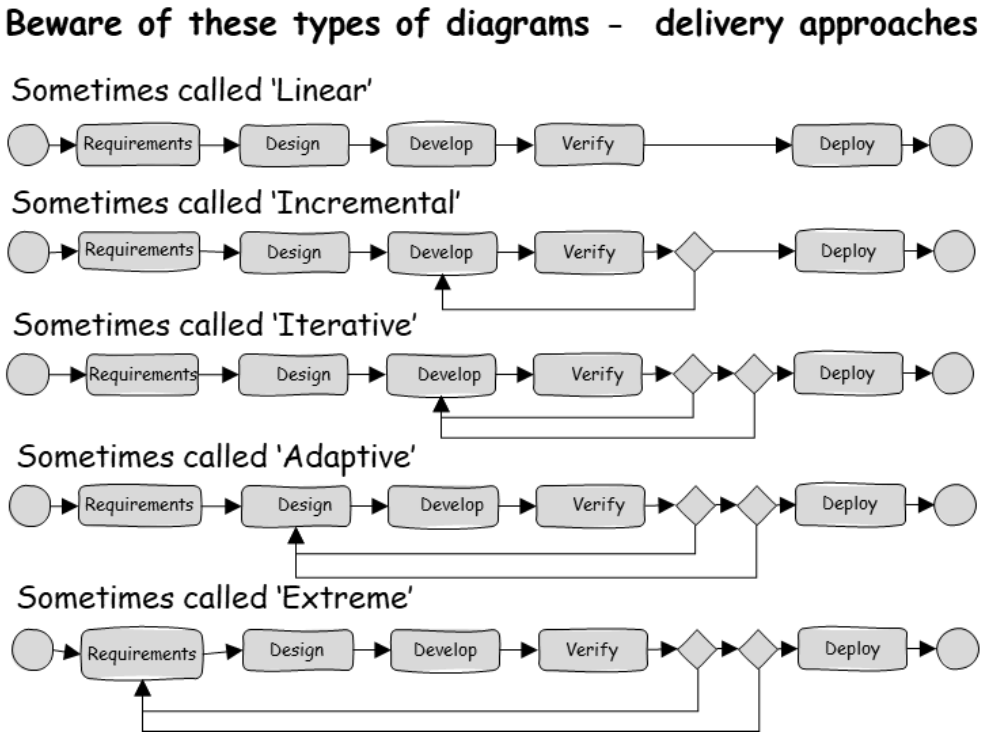


Figure 3 A typical delivery approaches seen in text books and academic papers, but often referred to, misleadingly, as 'life cycles'. Typical names of each model are shown. The names in the 'boxes' can differ considerably but the intent of the various authors is similar.

Examples similar to **Figure 3** can be found in Fernandez D J and Fernandez J D (2008) in their paper on 'agilism and traditional project management'. The base sequence they use is 'Scope', 'Design', 'Build', 'Test' and 'Deploy', which is software development oriented. However, their depiction is vague as to what their figure is meant to represent; the label at the top is "Life cycle, management approach, development approach or strategy?" but at the bottom is "Project management strategies based on complexity and uncertainty". If the definition of life cycle used in the standards is applied (see Example 1), Fernandez and Fernandez' models cannot be life cycles as they include loop backs to earlier 'boxes' and, in a life cycle, you cannot go back in time. They do, however, make sense if they are considered as processes.

Wysocki (2019) uses a similar set of diagrams and depictions in which he explains what he considers three types of project management life cycle models (traditional, agile, extreme) with examples of each. However, whilst his individual depictions almost mirror those in Fernandez and Fernandez and **Figure 3**, the names of his boxes appear to mirror PMI's former process groups[6] Further, he explicitly states the boxes are 'phases of a project life cycle'. PMI has always made it clear that their process groups were never meant to represent a project's phases. However, like Fernandez and Fernandez, Wysocki's diagrams make no sense as project life cycles, but do make sense if they are looked at as being processes.

[6] Scope (which PMI calls the Initiating Process Group); Plan (PMI's Planning Process Group; Launch (which PMI calls the Executing Process Group); Monitor and control (PMI's Monitoring and Controlling Process Group; Close (PMI's Closing Process Group)

There is no set definition for different approaches but in both Fernandez and Fernandez, and Wysocki's work the following types of models are described (referring to **Figure 3**):

- 'linear' (also called 'predictive' in some sources) is a simple sequence of activities.
- 'incremental' requires fixing the requirements and design in sequence, then developing part of the output, testing it and developing more of the output until it is completed, then deploying it in a single release.
- 'iterative' is the same as incremental except that the output is deployed in a number of releases.
- 'adaptive' is similar to iterative, except the design can change on each iteration.
- 'extreme' allows the requirements to change, but, strangely thereafter, design, development and verification are sequential as in the 'linear approach'.

The inference in both Fernandez and Fernandez, and Wysocki's models, as well as other sources that use this type of description, is that, iterative, adaptive and extreme are considered 'agile' and linear and incremental as 'traditional'.

Example 3 – Example of comparing non-comparable models

An example of a potentially confusing depiction of life cycles can be found in the current APM Body of Knowledge in their figure in section 1.2.2 Linear life cycles. In their figure, the picture on the left shows a life cycle as a Gannt chart (i.e. time bound) but the comparison on the right (from 1.2.3 'Iterative life cycles') is a representation of a process as it has iterations and does not comply with the term 'life cycle' as defined in the APM's own glossary (which is consistent with the definition in ISO 21502). The right-hand diagram is lifted from the Dynamic System Development Method (DSDM)[7]. By placing it in a chapter concerned with 'life cycles' and in the context of the diagram shown in its Figure 1.2.2, this might mislead a reader who is not familiar with the differences.

Separate project management from delivery approach

In a significant amount of the literature and 'chatter' on social media, the 'cycle' of life cycle, has been taken to mean 'iterative' as iterative processes can be depicted as 'cycles' as in the 'project control cycle'. Further, many examples cited tend to assume that there is only one type of output (usually software) and generally ignore the project life cycle as a means to bind together a

[7] Dynamic systems development method (DSDM) is an agile project delivery framework, initially used as a project managed context for software development. First released in 1994, DSDM originally sought to provide some discipline to the rapid application development (RAD) method. In later versions the DSDM Agile Project Framework was revised and became a generic approach to project management and solution delivery rather than being focused specifically on software development. The DSDM Agile Project Framework covers a wide range of activities across the whole project life cycle and includes strong foundations and governance, which set it apart from some other 'agile methods' and align it close to generic project management methods such as PRINCE2®. The DSDM Agile Project Framework is an iterative and incremental approach that embraces principles of agile development, including continuous user/customer involvement.

number of different delivery approaches for multiple outputs within the scope of a single project. It should be noted that each example in Fernandez and Fernandez, and Wysocki's models is only suitable for only one type of output. If a project has just one type of output the distinction between a 'life cycle' and a 'delivery approach' can become obscure. To the manager of that work, their delivery approach *is* their project management approach. Software developers often see it like this. However, if, as is normally the case, a project includes a number of different types of output, then a single project would have many types of output, each requiring a different delivery approach.

The terms' iterative' and 'incremental' are often seen, along with 'adaptive' as ways of adjusting a project's scope to manage complexity and reflect emerging requirements or changes in the project's context. However, 'iterative' and 'incremental' when used in relation to a project life cycle should be differentiated from 'iterative' and 'incremental' when used in relation to a delivery approach. Within a delivery approach, the output can be developed iteratively and either 'stored' for later release or released as soon as it is completed. Regardless of how an output is developed, it can be released for operations or put to use in many ways, as shown in the examples in **Figure 4**. It is therefore possible to have a project, which is adaptive, deploying its outputs in one release or a number of increments, where some of those outputs are produced using linear development approaches.

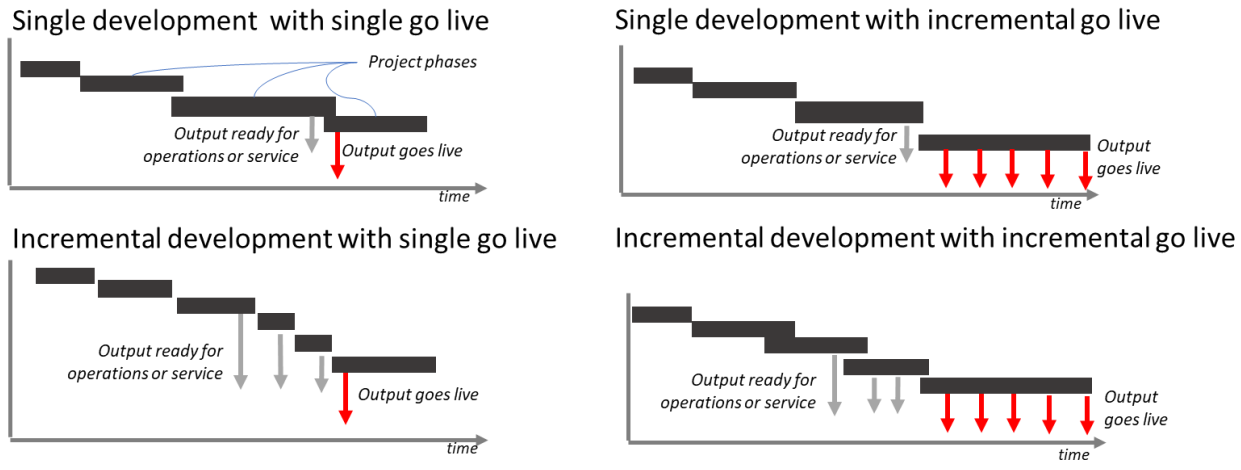


Figure 4 Some common options for life cycles for meeting the project's objectives

A way of comparing life cycles and delivery approaches

As the words used when discussing 'agile' cannot always be trusted, an approach is needed to verify the intent of the author, which is appropriate for any project, life cycle or delivery approach. As time is a feature of all projects, (they all have to start and end at some point in time), the Gannt chart can be used as a common way of comparing life cycles and delivery approaches. Even Scrum can be depicted in a Gannt chart, with dates for the start and end of each release and sprint, retrospective and other activities.

By representing the project in a Gannt chart it is easier, from a governance viewpoint, to see the delivery cadence, decision points (gates) and key activities. For iterative development, however it is not necessary to show the detail for each activity as working in iterative ways is too fluid to predict and track discrete activities; such detailed planning and tracking adds no value. It is only necessary to show an activity at a level which enables visibility and control and encompasses whatever part of the development process is being used. For example, a sprint, if using scrum, can be a single activity in a Gannt chart as, by definition, the timing, duration, resources and costs are fixed, with the scope (back log) varying. There is no need to add any more detail as the process used dictates what the team members do. A schedule plan needs only be developed at the level needed for control; this has always been considered good practice for iterative and repetitive activities, like trenching and pipelaying.

To illustrate this further, taking the example from the latest PMBOK® Guide, figure 2.10 (Life cycle with incremental development approach), each increment comprises 'plan, design and build' activities. There is also just one deployment towards the end. Figure 5 lays this out as a Gannt chart. No matter what delivery approach is chosen, the top-left to bottom-right, stair-case shape shown in the figure is always apparent. What distinguishes the delivery approach is the activities and deliverables undertaken in each phase. In this case PMI has shown a sequence of three predictive delivery approaches, and as a Gannt is used, it is possible to see how long each phase is planned to last.

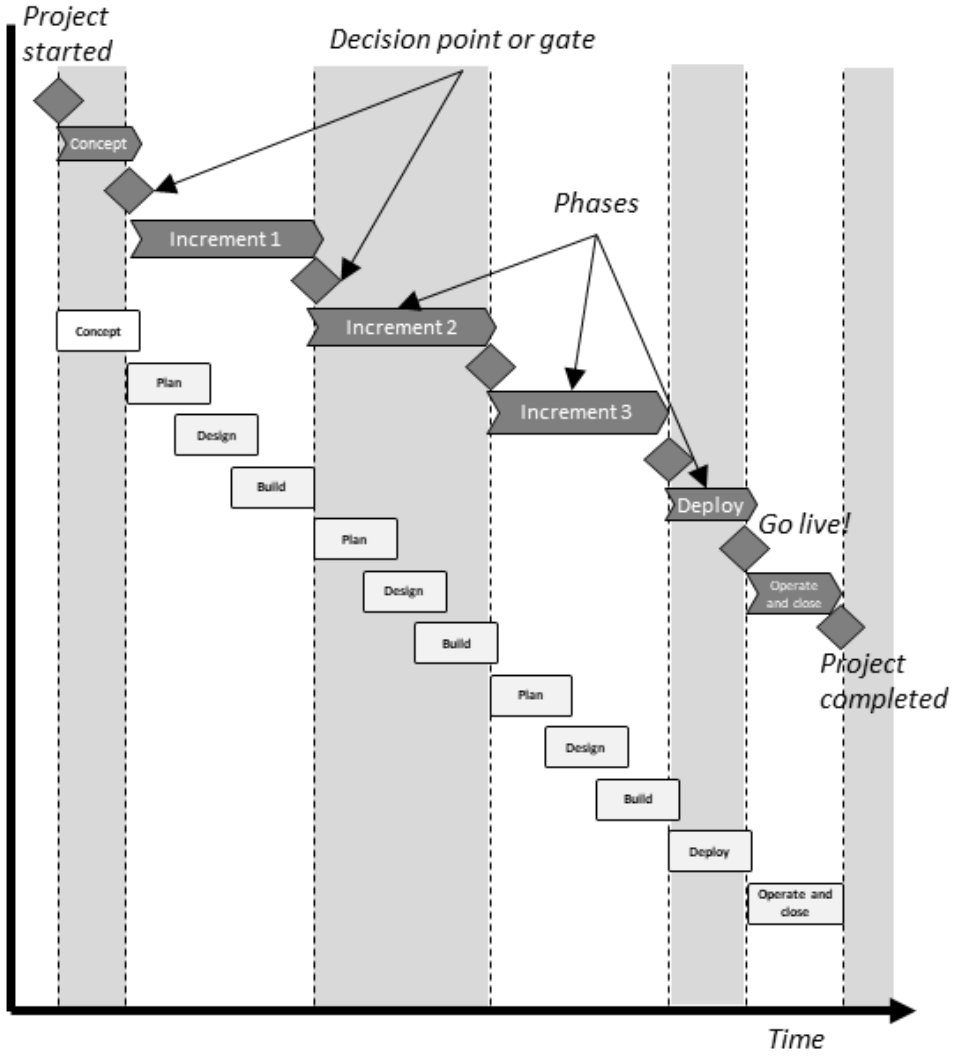


Figure 5 An example of an incremental delivery approach shown with appropriate project life cycle for one deployment and 'go live'

Figure 6 shows another example of an incremental development. In this example, each sprint is shown as a single activity to represent a scrum 'time box'. Unlike the example in **Figure 5**, deployment is incremental, with one deployment per phase rather than everything being deployed as a single output towards the end of the project. There is no need to detail everything that happens in each sprint as the process to be used defines that.

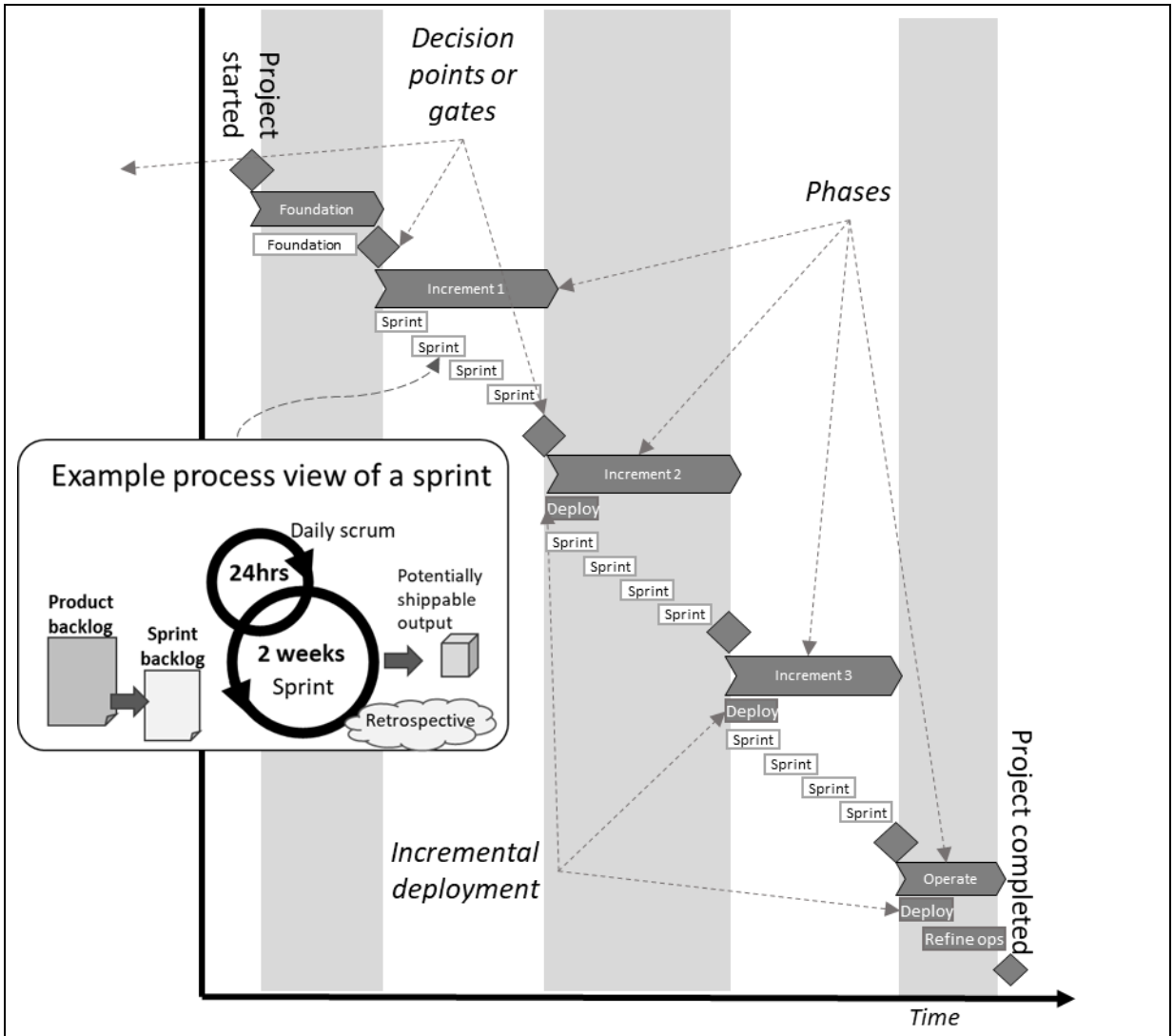


Figure 6 Example for a embedded in each increment and with incremental deployment and 'go lives'.

Is 'traditional' really 'traditional'?

There are so many ways 'agile' is described, and, although it is seldom defined, many writers compare it to what they term 'traditional project management'. As discussed earlier, 'agile' as generally presented, usually relates to a delivery approach and such comparisons are often flawed, like comparing apples with chairs (not even fruit or furniture) as in example earlier in this paper. Just as 'agile' and 'agile project management' are seldom defined, 'traditional' and 'traditional project management' are also rarely defined, but often used. 'Traditional project management' is, however, often equated to 'predictive' or 'linear' delivery approaches with 'waterfall' software development given as an example of such an approach. The inference of such an approach, is that requirements are fixed at the start, then the design is fixed to meet those requirements and the software built, with no option to change anything. If this is what

people refer to as 'traditional project management', it is a long way from long established project management practice, even though it might be a software development practice. For example, PRINCE2® has always been designed to enable a project to be adapted to reflect prevailing conditions and emerging requirements and issues. BS 6079 historically has taken a similar approach. The first edition of my book, *The Project Workout* was published in 1997 and is adaptive and for me to write about it, the practices must have been in use long before it was published. It seems, therefore, that what is 'traditional' from one person's perspective is a set of practices that don't apply in others' experience. The Oxford and Merriam Webster dictionaries have the following definitions of 'traditional' in **Table 2**

Table 2 Dictionary definitions of traditional

Oxford English Dictionary	Merriam Webster
Existing in or as part of a tradition; long-established. *//the traditional festivities of the Church year* *1.1Produced, done, or used in accordance with tradition.* *//a traditional fish soup* *1.2Habitually done, used, or found.* *//the traditional drinks in the clubhouse* *1.3(of a person or group) adhering to tradition, or to a particular tradition.* *//traditional Elgarians* *1.4(of jazz) in the style of the early 20th century.*	*of or relating to tradition: consisting of or derived from tradition* *//a traditional celebration* *2: handed down from age to age* *//traditional history* *//traditional songs/stories* *3: following or conforming to tradition: adhering to past practices or established conventions* //traditional morality //traditional values/beliefs //employing traditional methods of cooking

Looking at these definitions, as the 'Agile Manifesto' was published over 20 years ago in 2001, it could be considered long established and hence 'traditional'.

So what is 'agile'?

Context is important

Context is important, influencing how 'agile' is perceived and applied. If, for example it is applied in the context of the development of additional software features for an existing service, the mental picture is very different to when it is used in the context of developing a brand new digitally enabled service including people, processes, infrastructure and associated facilities. In the former, the 'agile' developments can be managed as a business-as-usual process rather than a project; Spotify is the most frequently given case study for this (Kniberg 2012).

Most people work within a defined area of expertise. Software developers do not generally end up managing the design and construction of a major rail route, such as the Elizabeth Line across London, though they could be involved in developing software in a work package or project for such a venture. Often they have no hand in the development of the servers and communications infrastructure their software uses nor in the products it is embedded within. For many software developers, the development of the software is 'their project', making the delivery approach and

project management approach indistinguishable from their perspective. Go up a few levels in the hierarchy and a project manager might see the development of the software as simply one of many work packages that need to be completed, each of which is aimed at developing a particular deliverable, using the most appropriate delivery approach.

Agile principles

As stated earlier, there is no generally recognised definition for 'agile' and yet many documents include or infer some commonalities. The European Union's PM^2 include principles which its authors consider are essential for 'being agile' (see **APPENDIX G: PM^2**). This is not the only document reviewed to include principles, but the only one specific to agility. Other authors have proposed agile principles, such as the Agile Alliance, but the Agile Alliance' principles are solely targeted at software development and as such, represent a delivery approach[8]. **Table 3** shows an example of how one of the documents reviewed in this paper maps to PM^2's agile principles. GovS 002 has been chosen as it is freely available to any reader of this paper.

Table 3 Example mapping to PM^2's agile principles

PM2 agile principle	References from GovS 002 and commentary
a focus on delivering value early on and frequently throughout a project	Value for money is a tenet throughout the standard, including in its purpose statement: …… ensuring value for money and the successful and timely delivery of government policy and business objectives, although there is no requirement for 'frequent delivery' as that is not always appropriate, nor is achieving it always proportionate.
decisions made based on what is known	Subclause 4.3 requires this but also says to take into account 'unknowns' as risks, when making a decision.
close cooperation among all parties involved	Principle 6 in clause 2 infers collaboration across members of the team.
continuous stakeholder involvement at all levels	Subclause 7.12 says that stakeholders should be identified, and their interests and expectations understood and represented. Further, subclause 8.6 says validation should be continuous throughout the life cycle and may be iterative in nature with the requirements, design and solution evolving as work progresses.
involving team members in planning	Subclause 7.2.2: Planning should be a collaborative activity involving team members advising on the planning of their work.
incremental development with short cycles	Subclauses 8.2 to 8.6 state that design should be in accordance with a defined approach and may be predictive, incremental, iterative, adaptive or hybrid, including agile approaches and that requirements can be modified as design progresses, with ongoing verification and validation. GovS 002 does not have any requirement for short cycles, leaving the choice of delivery approach and design of the life cycle to be 'appropriate and proportionate.'

[8] Principle 3 in the Agile Alliance's principles is: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. Principle 7 is: Working software is the primary measure of progress. The principles for the original Agile Manifest are also software related.

scope management through the continuous (re)prioritisation of tasks	Subclause 8.3 says requirements should be uniquely identifiable, current, mutually consistent, understandable, unambiguous, prioritised and validated. Subclause 7.7 explains that any changes need to be justified, whether within a phase to at the start of a new phase. Subclause 7.2.2 says s cope may be refined and clarified as work progresses.
embracing change, continuous learning and improvement	Subclause 7.2.2: Planning may be progressive through the life cycle. Subclause 8.8, learning from experience
just enough documentation and control.	Principle 3 in clause 2 says governance and management frameworks, and controls are proportionate and appropriate to the work and the level of prevailing risk. Too much documentation would be 'inappropriate' and not proportionate.

PRINCE2® Agile uses the same seven principles[9] as the primary PRINCE2® manual. This makes sense, as a 'principle' is a fundamental truth or proposition that serves as the foundation for a system of belief or behaviour or for a chain of reasoning. Being fundamental, a principle applies in all circumstances[10]. Importantly, PRINCE2® Agile takes the view that there is no such thing as an 'agile project'; agile is not binary and agile frameworks, techniques and behaviours can be applied to any project or part of a project.

Agile behaviours and mindset

Generally, the source documents cited in this paper, whilst emphasising behaviours are important, do not state what those behaviours are, although IPMA's ICB4 does concentrate on the competencies required. Behaviours are either outside the scope of the document, such as in the ISO 21500 series, or inferred in the narrative. As noted at the start of this paper, the fact that the standards do not prescribe behaviours is understandable as behaviours often relate to culture which differs considerably across the world, often influenced by both the social norms and political environment. They do, however state that behaviours are an aspect of competency and an important factor in the management of work and the resulting outcomes, especially in relation to business and social change.

PRINCE2® Agile, however, lists five behaviours it considers prerequisite for work to be done in an 'agile' way. These are:

- **transparency,** as this enables speed, clarity and engagement, even if the news is not good.
- **collaboration,** as a motivated and respectful team is greater than the sum of its parts when people work together and provide cover for one another.
- **rich communication,** where information passes freely in a culture of commitment and trust.

[9] The PRINCE2® principles are: continued business justification; learn from experience; defined roles and responsibilities; manage by stages; manage by exception; focus on product; tailor to suit the project environment/
[10] The principles in BS 6079 and GovS 002 meet this test.

- **self-organization,** as the people closest to the work usually know best how to get the job done and should be trusted to do it.
- **exploration,** as projects are difficult, and in order to create 'the right thing', 'the right thing' needs to be defined and this needs people being given the opportunity to investigate, learn and learn again.

It should be noted that none of these behaviours is unique to agile approaches, it is simply that these are emphasised in most or all agile approaches. Further, none of them breaches, but rather, in some cases, directly support the overriding principles in the documents cited in this paper.

A fully flexible life cycle and delivery approach?

The delivery approaches shown in Figure 3 are ones that the author has rarely seen used in practice, except on single, 'one output' projects or work packages. Experience on larger scale programmes, projects and work packages has been that more flexibility than implied in any of the models is needed. No matter what phase of a project is under way, issues might be found which require the design of a component of the solution to be revisited, which might in turn lead one to question the requirements, leading to adjustments across the whole of the solution. This leads to rework, either within the current phase or in later phase. Such an issue could be the result of changing circumstances (the world does not stop for anyone's project) or could be the result of new information or the discovery of a defect[11]. Accordingly, any aspect of the solution might need addressing at any point in time. This requires a level of 'agility' beyond what is shown in any of the models in **Figure 3**, where any activity can be triggered from any other activity and, if necessary ripple back through the sequence to where the issue can be resolved, as shown in the example in **Figure 7**. In extreme circumstances this could also lead to a review of the fundamental objectives for the project.

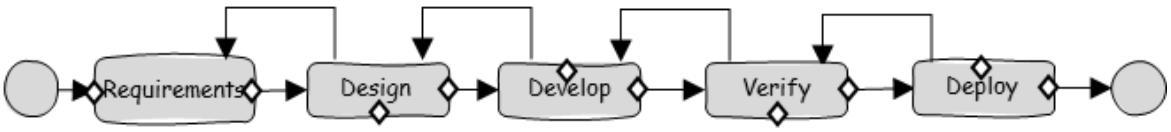


Figure 7 A fully flexible delivery approach, with multiple feedback loops

As noted earlier, most of the models proposed to illustrate 'agile' working are focussed on a single output. In practice, multiple outputs are needed, even for seemingly simple products. For example, software needs to run on hardware, data used by the software needs communications infrastructure and security protocols, the users need an interface and suitable devices, training applications, environments and data are needed, data might have to be migrated from legacy systems, requiring special applications and environments. All this without even considering

[11] In this context a defect is an identified error within an approved deliverable and can be applied to a y deliverable, whether interim or final. (Buttrick 2020).

where these are physically: on a plane, train, boat, car, house, in a tunnel, on a road, in an office or space vehicle? Buttrick (2020) proposes an indicative model (see **Figure 8**) which elaborates **Figure 7**, for a fully flexible development approach, comprising an overall system (or solution) and any number of sub-systems (solution components). Whilst it appears sequential, like a waterfall development approach, the 'rework' loops shows that it is neither a life cycle nor 'waterfall'. As a delivery approach, there is no clue in **Figure 8** as to when these practices take place.

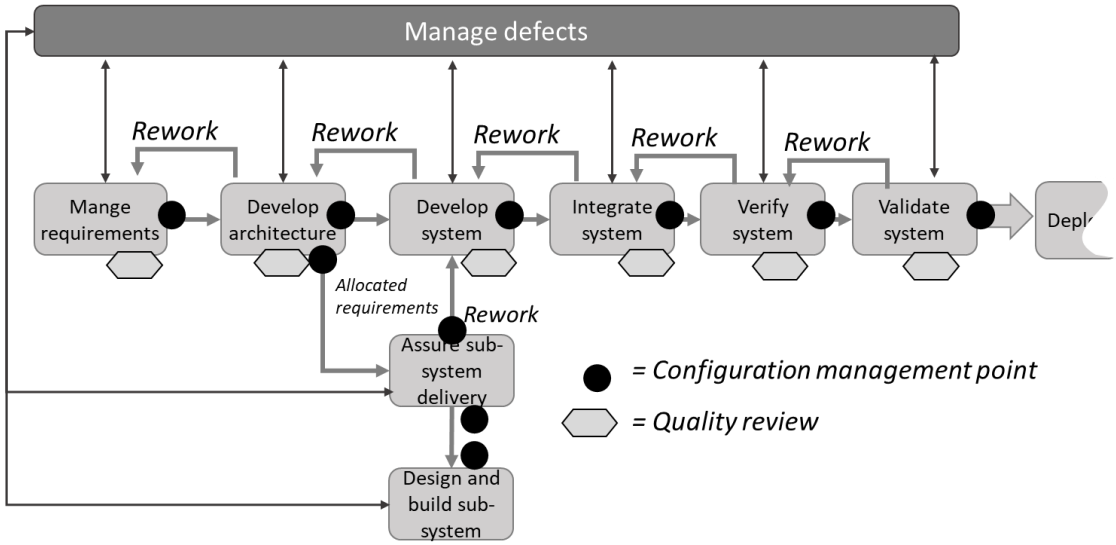


Figure 8 An example solution development process[12]

Figure 9, however, shows these delivery practices in relation to the project life cycle, with an indicative graph showing when and how often each practice is used. For example, 'Requirements' is used predominately at the front end, but is also drawn on throughout the rest of the life cycle, if needed. Validation closely follows 'Develop, integrate and verify system' and predominately happens in the middle part of the life cycle. 'Develop', 'Integrate' and 'verify' have been merged for simplicity in the diagram on the basis that, in this case, integration and verification are continuous throughout development. 'Validation' is predominately at the back end but is used to validate the requirements and design early on as well.

[12] Source: Figure 24.6, Buttrick, R. The Programme and Portfolio Workout, 1st edition, Routledge, 2020

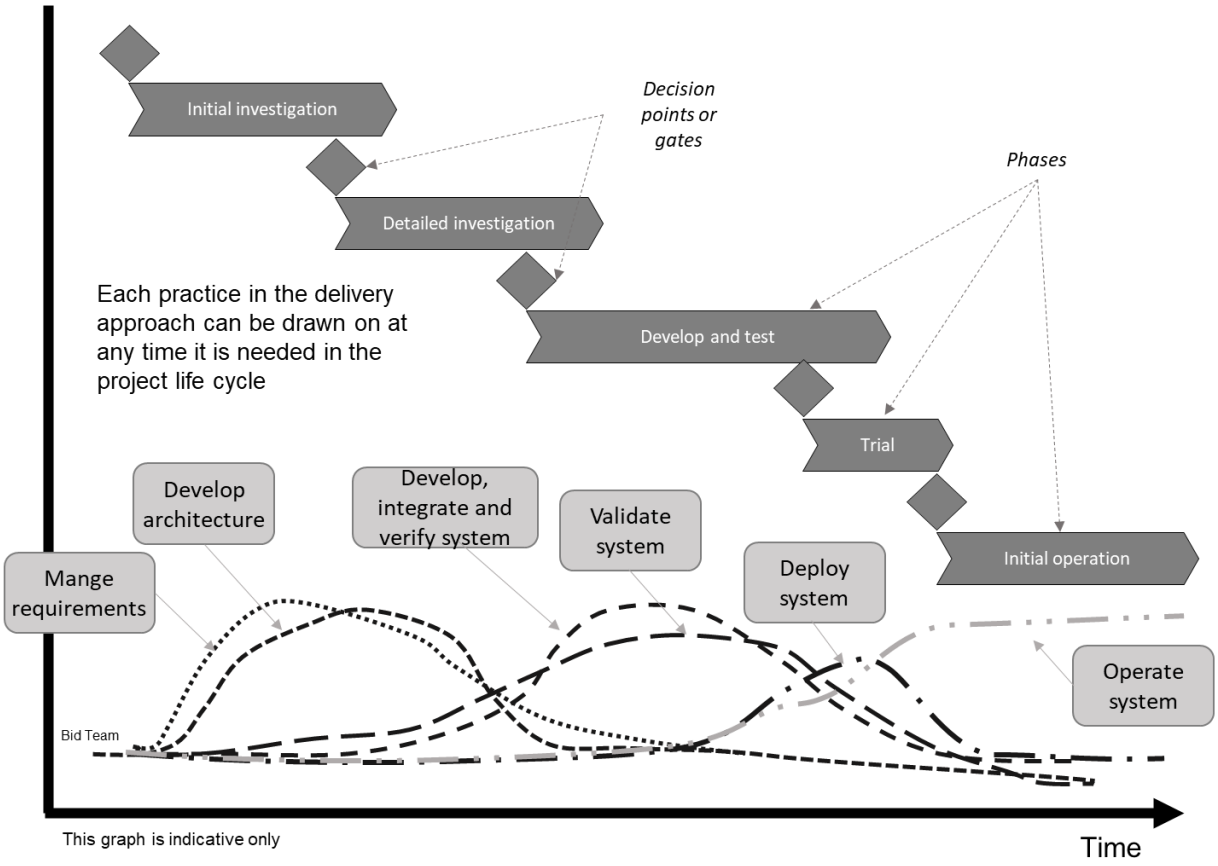


Figure 9 Indicative relationship between a project life cycle and the practices in a delivery approach

For a more in-depth discussion on project life cycles, see Buttrick (2019,2019a and 2020).

Conclusion and recommendations

All the primary documents cited in this paper are inclusive of 'agile' working

Whether one considers 'agile' to relate to a management framework, behaviour or technique, all the primary documents reviewed in this paper are intentionally inclusive of 'agile' approaches. They might not use terms which are commonly seen in some published agile approaches, just as they do not include specific terms from any other delivery approaches. The intent is that the documents are independent and inclusive of any delivery approach, whether current or in the future.

Do not use the terms 'traditional project' or 'agile project'

Do not use the terms 'traditional project' or 'agile project' as these terms have no common meaning and can be a source of antagonism. One person's 'traditional' is an anathema to

another, and there are so many views on these undefined terms, it is best not to use them at all. Similarly, avoid using 'traditional project management' and 'agile project management'. Instead describe what you mean, using plain language and appropriate diagrams.

Behaviours and mindset are important – do not ignore this aspect

Behaviours and mindset are important when considering project management and delivery approaches. The right behaviours can make a particular approach or method work superbly, but the wrong ones can cause it to fail. Behaviours and mindset should match the methods, processes and techniques used as well as the culture of the organisations involved in the work and the social norms of wider society within which the work is undertaken.

Treat life cycle as distinct from delivery approach

Keep life cycle distinct from delivery approaches; both are needed. ISO 21502 clearly distinguishes a project life cycle from a delivery approach and recommends the relationship between these two entities. This distinction is echoed in the primary publications cited in this paper. As such, these documents are inclusive of any type of delivery approach, existing or in the future. Having decided on the appropriate life cycle and delivery approaches, make sure that fit together neatly.

Use the Gannt chart to compare life cycles and approaches

As different people use the same words in different ways, a reliable way to compare different strategies for undertaking a project, in terms of the life cycle and delivery approaches, is needed. As it is important how long something takes, the Gannt chart is a useful tool for comparing management frameworks as it can show the phases, key activities and milestones, as well as the delivery cadence. This doesn't mean the Gannt chart has to be used for tracking progress; whatever tracking technique is appropriate should be used.

Don't be bogged down in fruitless arguments

If social media posts are representative of background discussions on the topic of agile and agility, emotions seem to run high. Don't be drawn into binary arguments (such as agile = good/bad), but rather look at different viewpoints from the perspective of those making statements you disagree with. What they say could be correct in their own context and day to day lives but might not fit your circumstances. So, keep focussed on your own situation and build agility into your life cycles and delivery approaches which meet your needs, not necessarily someone else's in a different organisation working in a totally different context. Do however be mindful that there are likely to be people in your teams who have different opinions; it has always been the case that project managers and other leaders need to engage their people and form effective teams to deal with differing opinions.

APPENDIX A: Management frameworks, techniques, and behaviours and mindsets

Table 4 Management frameworks, techniques, and behaviours and mindsets

Aspect	Comment on the various sources
Management frameworks	ISO 21502, ISO 21503, ISO 21504 use the term 'framework' as a collective for providing direction and working methods to individuals involved in the scope of the work the framework applies to. In effect, it changes the 'why' and the 'what' defined in the standards into 'how' to do it in practice, including, but not limited to, processes, methods and techniques. These standards are deliberately independent of the delivery approaches used so that their application can be as wide as possible. BS6079, GovS 002 and ANSI/PMI 99-001-2021 are standards and take a similar approach to the ISO 21500 series, being independent of 'how' work is done. PM2 and the AXELOS sources are methods and come under the umbrella of a management framework. IPMA's and GAPP's publications are competency based and can be considered part of a management framework. The bodies of knowledge seek to be independent of the delivery approaches used and form a curriculum for professional competence.
Techniques	The ISO 21500 series of international standards do not define what techniques should be used. Techniques are considered a part of a management framework and as such deal with 'how' to put the 'why and what' (as defined in the standards) into practice. Only the AXELOS and PM2 methods describe techniques to be used but all the sources promote tailoring so that work is done in a way which is appropriate and proportionate to the need. None of the sources cover how specific types of deliverable should be developed
Behaviour and mindset	The ISO 21500 series international standards do not define what behaviours are effective or should be used. Behaviours are often a factor relating to culture which differs considerably across the world, often influenced by both the social norms and political environment. The standards do, however, state that behaviours are an aspect of competency and a factor in the management of work and the resulting outcomes, especially in relation to business and social change. The competency baselines and bodies of knowledge seek to highlight the skills and competencies, and by inference, the type of behaviours to be encouraged. The bodies of knowledge also include some advice on this. Specific competencies, and the appropriate behaviours, are often required for specific methods and are not necessarily generic..

APPENDIX B: Terminology

Examples of use of terminology are given in the following tables.

Table 5 Example dictionary definitions

Term	Relevant Oxford English Dictionary definition
agile	Business. Of a company, business activity, product, etc.: able to change or be changed rapidly in response to customer needs and market forces; adaptable, flexible, responsive.
adaptive	Having a capacity for or tendency towards adaptation; spec. able to adjust to new situations or surroundings;
hybrid	Anything derived from heterogeneous sources, or composed of different or incongruous elements; *Computing*. Utilizing or involving both analogue and digital methods.
incremental	Of or relating to an increment or increments (usually in the mathematical sense (incrementally: by increments; in small successive stages)
iterative	Characterized by repeating or being repeated.
lean	Poor or meagre in quantity or quality; slight, mean; *NOTE: Merriam Webster also has: characterized by economy*
linear	Designating or pertaining to programmed learning aimed at step-by-step progress in which the material is broken down into small steps each of which must elicit a correct response before the next one is presented; frequently contrasted with branching methods. *NOTE: the above definition relates to education but most closely reflects its usage in the context of its uses when juxtaposed with 'agile'*.
predictive	That has the character, quality, or function of predicting the future; prophetic; that is a predictor of a future event, circumstance, etc
waterfall	Designating an approach to project management (esp. in software development) employing sequential stages, having little scope to react to changing requirements. Also: of or relating to this approach. Frequently contrasted with agile adj. 5. Contrasted with agile

Table 6 Examples of terminology

Established term	'Agile' equivalents	Comment
business case	roadmap	The ISO 21500 series, BS6079 and GovS 002 do not define the name of any management deliverables but does require the work to be justified.
requirements	epic; user stories; backlog item	User stories relate to a particular way of defining requirements from a user's perspective which does not suit itself to defining every type of requirement required from a whole system perspective.
scope	backlog	Not a full equivalent as the scope' is generally everything that needs to be done, whereas the backlog is what is left to be done, but might not be completed beyond the minimum viable product.
lessons learned	retrospectives ; plus/delta	The emphasis in many approaches is that lessons are captured as soon as they are recognised and not just as set piece meetings at the end of a piece of work (sprint, work package, phase etc.). At such meetings particular techniques are often used to solicit lessons and gain further insight.
gate	demonstrations	Gates serve more than providing a demonstration of what has been achieved as gates are focussed on deciding what (if anything) to do next

phase	increment	
report	dashboard; information radiator	The ISO 21500 series, BS6079 and GovS 002 do not prescribe the format, style, frequency etc of reporting but what needs to be reported.
team	tribe; teamlet; swarm	Teamlets and swarms are sub-teams.
work package	sprint	Sprints have a fixed schedule, resources and cost baseline and are always completed to plan. Unfinished work is moved to the next sprint, whereas a work package often has a defined scope which needs to be completed.
meeting	daily stand-up; huddle	All approaches require meetings but some 'agile' approaches require the team to meet daily.
sponsor	big boss; executive sponsor	
iteration	Timebox; sprint	In some 'agile' approaches, iterations are timeboxed and called sprints with each sprint of the same duration.

APPENDIX C: The ISO 21500 series of international standards

The ISO 21500 series of standards cover portfolio, programme and project management, together with governance and some other special subjects. ISO 21502 replaced a process-based ISO 21500 which was very similar to the prevailing ANSI standard at that time, covering only the project manager's role. This has since been replaced by ISO 21502, which is a value driven approach to project management and encompasses many more roles from the sponsoring organisation to the team members.

ISO 21500:2020: ISO 21500 provides an overview of the other standards in the series and makes no specific mention of agile, predictive, adaptive or other approaches

ISO 21502:2020, project management: ISO 21502 is focussed on project management and can encompass any delivery approach used (see clause 1). ISO 21502 does not restrict the roles to only those described in the standard, allowing roles to be defined to suit the work being done (see subclause 4.5.11).
Whilst ISO 21502 advocates a phased approach to undertaking a project it does not define the number, names or nature of the individual phases (see subclause 4.4). The nature of the decision points (gates) and phases is also defined and should include specific milestones to be reflected in the schedule plan (see also subclause 4.4). The definition of the actual delivery approaches is outside the scope of the standard. This is so that the standard does not constrain the use of new or novel approaches, whether existing at the time the standard was written or invented after the standard was published. This is emphasised in Figure 4 of the standard. ISO 21502 recommends that the governance and management framework for a project is defined and that this includes the approaches for delivering the project's outputs (i.e. delivery approach). See subclause 6.5.3. Clause 6.7 then emphasises the need for tailoring. By including this in subclause 6.7, ISO 21502 relates delivery approaches to work happening within a work package, which is what is shown in the standard's Figure 4. The standard, in subclause 7.2, emphasises that the approach to planning (including scheduling) should be suited to the delivery approach used.

ISO 21503:2022, programme management: ISO 21503 is concerned with programme management. It can be used with any delivery approach and includes no specific mention of agile, predictive or adaptive approaches.

ISO 21504:2022, portfolio management: ISO 21504 is concerned with portfolio management. It can be used with any delivery approach and includes no specific mention of agile, predictive or adaptive approaches.

ISO 21505:2017, governance: ISO 21505 is concerned with governance for projects, programmes and portfolios. It can be used with any delivery approach and includes no specific mention of agile, predictive or adaptive approaches.

APPENDIX D: BS 6079:2019, project management

BS 6079 is the United Kingdom's official standard on project management. Whist BSI has also adopted ISO 21502, BS6079 has been retained as it includes aspects considered important, but which are not included in ISO 21502. It is a principles-based standard (like its 2010 predecessor) with practices built off the principles.

Whilst it is not explicit in its introduction, BS 6079 implicitly allows any type of delivery approach. BS6079 includes the definition of the delivery approach as an element of project management see subclause 4.2.3. BS 6079, then explains more about 'delivery approaches' in clause 4.5. The 'Managing needs and requirements' subclause 13.3.2 explicitly mentions iterative delivery approaches and by implication, the 'user story' approach to requirements used in many agile methods. The planning clause 13.1.1.2 explicitly allows methods where the scope is fixed or where scope can be variable, such as in some agile approaches; this is elaborated in the standard's Appendix A.

APPENDIX E: GovS 002, project delivery

GovS 002 is the United Kingdom government's internal functional standard, and its use is mandatory for public bodies. It covers portfolio, programme and project management. It is a principles-based standard with practices built off those principles.

This project delivery standard defines, in subclause 3.3, the practices needed for delivering successful outcomes. Like the ISO 21500 series of standards, it defines the 'why' and the 'what' for each but does not define 'how' work is to be undertaken. This is a similar approach taken to life cycles in BS 6079 and ISO 21502. It explains this in subclause 2.1. The context, clause 3, especially deals with and explains the relationship between delivery approaches and the content of the standard with an explanatory diagram. Clause 5.2 on portfolio management makes it clear that portfolio management is both iterative and adaptive. Subclause 7.2.2 on planning emphasises the iterative and progressive nature of planning.

Clause 8.2 on quality emphasises, in a note, that the delivery approach needs to be chosen carefully and cites 'agile' as an example. Clause 8.3, on requirements, states that requirements do not have to be confirmed at the start and can evolve with the design and development. It also cites agile jargon as example of different names for the plain English 'requirement'. Clause 8.4 on design is allowed to take any appropriate form and thereby implies that the waterfall software development method (requirements, design, build, test, deploy) is not the recommended approach. Clause 8.6 on verification and validation simply states that verification (such as testing) and validation (such as trials) should not be left to the end of the project but happen throughout the life cycle.

APPENDIX F: ANSI/PMI 99-001-2021, project management

ANSI/PMI 99-001-2021 is the USA's official standard on project management, which is published as Part 1, *The Standard for Project Management* together with Part 2 PMI's PMBOK® Guide, the *Guide to the project management body of knowledge*, 7th edition, (see Appendix I). The ANSI standard describes itself as 'principles based' and has 12 principles. Unlike the other standards cited in this paper, the ANSI standard is mostly descriptive in nature and contains no practices to support its principles. The ANSI standard is very light on recommendations (only 23 recommendations ('should' statements) compared to ISO 21502, which has 366).

Clause 1.1, like the ISO, BS and GovS standards emphasizes the inclusivity or all delivery approaches. Unlike them, the ANSI standard only recognises one form of what the ISO series calls 'other related work', being 'Operations'. The implication is also that portfolios, programmes and projects do not include operations; it is therefore more limited in its application and scope than the other standards. The ANSI standard's intent to encompass any delivery approach is further underlined in their terms and definitions in clause 1.2. Clause 2.3.2 on presenting objectives, the need for feedback is emphasised although there is no specific recommendation in the standard. Clause 2.3.6 on providing business direction, it implies such direction should relate to a defined cadence or checkpoint. In the other standards these are advised to be continuous. Clause 3.4 focus on value picks out adaptive approaches for working with customers to determine what is of most value to them. No recommendations are given. The ANSI standard follows none of the generally accepted norms for writing standards.

APPENDIX G: PM², project management

The European Union's PM² method discusses 'agile' in clause 3.7, describing it as an approach to managing projects, based on a specific set of principles. It lists those principles and is the only document to do so, and so defines a set of criteria to be met if work is to be considered 'agile'. These principles are:

- a focus on delivering value early on and frequently throughout a project
- decisions made based on what is known
- close cooperation among all parties involved
- continuous stakeholder involvement at all levels
- involving team members in planning
- incremental development with short cycles
- scope management through the continuous (re)prioritisation of tasks
- embracing change, continuous learning and improvement
- just enough documentation and control.

As PM² can be used with any delivery approach it also states that the method can be used as a structure to help achieve 'agility' whilst still accommodating tight procurement and audit requirements, which are a necessary feature of public sector projects.

APPENDIX H: IPMA COMPETENCY BASELINES

IPMA, PEB

In subclause A.2c, IPMA's PEB contains no advice and does not appear to distinguish between project life cycle and delivery approaches, but it does explicitly acknowledge 'agile', along with other approaches, as a management strategy. The document however recognises the project life cycle in a similar way to ISO 21502 by describing, in subclause 6.2, that a project has certain phases, and that a project life cycle may differ in both the number of phases and the detail within each of these phases.

IPMA Individual Competence Baseline for project management v4.0 2015

The individual competency framework does call for individuals to have knowledge of 'agile development' in relation to skills for managing time and scope (see subclause 4.5.3 for time, and subclause 4.5.4 for scope).

IPMA Organisational Competence Baseline (OCB)

Whilst IPMA's organisational competence baseline does not explicitly mention 'agile', 'iterative' nor 'incremental' it does, in section 8, differentiate between delivery approach (it calls them 'delivery processes') and programme and project management. Further it stresses they need to work together.

IPMA ICB4 in an agile world

IPMA has published a supplement to ICB4 which specifically deals with 'agile'. This document is unique in relation to the others which have been published in that it relates to competencies considered to be 'good' to apply generally and to apply to 'strategy' and 'practice'. Other publications, like PRINCE2® and PRINCE2® Agile also deal with behaviours but as a thread running through the document, not as the sole purpose of the document. In effect, this IPMA document recommends the competencies for undertaking projects with 'agility'. IPMA does not define 'agile' or its derivatives, that needs to be deduced from the content. The document covers 23 competencies, 5 from a specific perspective (such as strategy and governance), 10 on people (such as self-reflection, leadership and resourcefulness) and 13 on practices (such as design, scope and time).

Many of these are not new, nor unique to 'agile' and have been a stable part of leadership development and training for many years. The question is why they seem to be only applicable to 'agile work' when many have been applicable to projects and other business activities for a long time? If the 'agile' word was taken out, the competencies would still hold true, except for the few that define how something has to be done, such as using an 'information radiator' or having a 'retrospective.

APPENDIX I: APMBoK, portfolio, programme and project management

The APMBoK is the UK's chartered Association for Project Management's key reference document and describes itself as a set of concepts, terms and activities that make up a professional domain.

The APM Bok addresses 'agile' approaches and development in its life cycle chapter (1.2.3) which covers iterative approaches and time boxing in particular. It goes on to discuss iterative or agile methods but, as noted in this paper, it risks confusing life cycles and delivery approaches. In section 1.2. 4 it discusses 'hybrid' life cycle which is says is a fusion of predictive and iterative approaches, with the iterative approaches often used in early requirements gathering.

APPENDIX J: PMBOK® Guide, 7th edition, project management

The PMBOK® Guide contains eight 'performance domains' which are a group of related activities that PMI considers critical for the effective delivery of a project's outcomes. These are: Stakeholders, Team, Development Approach and Life Cycle, Planning, Project Work, Delivery, Measurement and Uncertainty. The use of these is intended to be guided by the ANSI standard (see Appendix E). In addition, there is a section on tailoring and another on commonly used techniques. The back cover of the PMBoK® Guide makes it clear that the guide reflects the full range of delivery approaches (predictive, traditional, adaptive, agile, hybrid etc), although 'traditional' is not defined. The PMBOK® Guide, part 2, clearly treats project life cycle and delivery approach as separate entities. This is emphasised in its glossary of definitions, and it devotes a complete 'performance domain' (section 2.3) to delivery approach, project life cycle and the relationship between them. In this respect, the PMBOK® Guide uses the same approach as ISO 21502. It then goes on in subclauses 2.3.1 and 2.3.3, to explain the way they fit together and the almost limitless permutations that can result. It uses, in Figure 2-7, almost the same diagram as used in APM BoK but with a significant change: the reference is to 'development approaches' not to 'life cycle'. By doing this they make the distinction explicit. This distinction is further emphasised in a box out in subclause 2.3.7, presumably because the confusion in peoples' minds between 'life cycle' and 'development approach' (delivery approaches) is at the source of so much fruitless argument and disagreement over concepts which are not comparable.

Like PRINCE2®, the PMBOK® Guide now devotes a lot of content to tailoring. The tailoring section in the PMBOK® Guide provides guidance on the appropriate and proportionate application of project management practices and, as such, provides advice behind ISO 21502's subclause 6.5.3 on the project governance and management approach. The PMBOK® Guide describes, in subclause 3.3.1, how a project can include a number of different outputs, each of which might require a different delivery approach, but are bounded within the same phase of the project. The PMBOK® Guide then goes on, in subclause 3.4.1, to provide guidance on selecting the delivery approach(es).

APPENDIX K: PRINCE2® and PRINCE2® Agile, project management

PRINCE2®is a principles-based method. PRINCE2®'s seven principles must always be complied with and are supported by seven themes (Business case, Organisation, Quality, Plans, Risk, Change, Progress) which are equivalent to ISO 21502's 'Management practices for projects' and seven processes (covering the sponsor, project manager and work package manager's accountabilities) which are equivalent to ISO 21502's 'Integrated project management practices'. It also makes it clear that the specialists should use their own working practices for the development of the outputs and the project manager should ensure the specialist work packages are an integral part of the project (see section 4.3.3). An example of the focus PRINCE2®puts on agile is that it is included as one of commonly used tailoring needs (see 4.3.4.2). The others are: simple projects, projects with a commercial customer and supplier relationship, projects involving multiple organisations and projects within programmes. It does the same for tailoring the themes (see section 5.2). PRINCE2® Agile has the same structure, principles, processes and themes as the primary document but illustrates how agile management frameworks, techniques and behaviours can be used in that context.

References for further reading

Agile Alliance, The 12 Principles behind the Agile Manifesto (viewed 2022)

Agile Manifesto [https://agilemanifesto.org/] (viewed 2022)

ANSI/PMI 99-001-2021, The Standard for Project Management, PMI, 2021

APMBoK, APM Body of Knowledge, 7th edition, 2019

Bennet N, Buttrick R, Stanton P, PRINCE2®, Projects in a controlled environment, 6th edition, AXELOS, 2017

Buttrick R, Project lifecycles and PRINCE2®, AXELOS, 2019

BS 6079:2019. Project management – Principles and guidance for the management of projects, BSI, 2019

Buttrick, R. The Project Workout, 5th edition. Routledge, 2019

Buttrick, R. (2019a). The project framework: understanding gates and stages, PM World Journal, Vol. VIII, Issue I (January). https://pmworldlibrary.net/wp-content/uploads/2019/01/pmwj78-Jan2019-Buttrick-project-framework-understanding-gates-and-stages.pdf

Buttrick, R. The Programme and Portfolio Workout, 1st edition, Routledge, 2020

Fernandez DJ and Fernandez JD, Agile project management - agilism versus traditional approaches, Journal of Computer Information Systems 49(2). 2008

GovS 002, Project delivery functional standard, v2, HM Government, 2020.

IPMA standards –Baseline (PEB), V1, 2016

IPMA standards –Baseline (ICB), V4, 2015

IPMA standards –Baseline (OCB), V1.1, 2016

IPMA standards –Baseline (ICB4 in an agile world), V1, 2018

ISO 21500:2020, Project, programme and portfolio management — Context and concepts, ISO, 2020

ISO 21502:2020, Project, programme and portfolio management — Guidance on project management, ISO, 2020

ISO 21503:2022, Project, programme and portfolio management — Guidance on programme management, ISO, 2022

ISO 21504:2022, Project, programme and portfolio management — Guidance on portfolio management, ISO, 2022

ISO 21505:2017, Project, programme and portfolio management — Guidance on governance, ISO, 2017

Kniberg H and Ivarsson A, Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds, 2012

PM^{2,} Project Management Methodology, Guide 3.0, 2018

PM^2 Agile, v3.01, 2021

PMBOK® Guide, A Guide to the Project Management Body of knowledge, 7th edition, PMI, 2021

Richards, K, PRINCE2® Agile, 1st edition, AXELOS, 2015

Wysocki Robert K, Effective Project Management, Traditional, Agile, Extreme, 8th edition (2019), Wiley

About this article

This article builds on concepts in *The Programme and Portfolio Workout* and *The Project Workout* which provide practical advice and techniques to direct and manage portfolios, programmes and projects in a structured, yet agile, way. The article takes Chapter 13 from *The Project Workout* as a starting point and goes into greater depth on the challenges of using, or even discussing, 'agile' in the context of projects. It then incorporates the concepts in Chapter 24 of *The Programme and Portfolio Workout* with respect to understanding delivery approaches in the context of project life cycles.

About the Author

Robert Buttrick

United Kingdom

Robert Buttrick is an independent advisor on portfolio, programme and project management, specialising in business-driven methods, processes and standards. Recent clients include the UK's Cabinet Office, Network Rail, and AXELOS. He is a Visiting Teaching Fellow at the University of Warwick, a member of the British Standards Institute's committee MS2 for project management and is a UK Principal Expert on the equivalent ISO technical committee, TC258 (dealing with international standards on portfolio, programme and project management.)

As well as being the author of "*The Programme and Portfolio Workout*" and the "*The Project Workout",* Robert has worked in one of the world's most turbulent and challenging industrial sectors, telecommunications, where he has been accountable for creating and running project-based frameworks for managing change, involving the direction of portfolios of over 2500 projects, totalling £4bn spend per year. Before this, Robert was with PA Consulting, a management and technology consultancy. There, he specialised in business-led project management, advising clients such as TSB Bank, National Rivers Authority, Property Services Agency, Avon Industrial Polymers, National Westminster Bank and RHM.

After graduating from the University of Liverpool with a first class honours degree, he joined Sir Alexander Gibb & Partners (now Jacobs) who provided consulting, design and management services for infrastructure, working in countries as diverse as Kenya, Mauritius, Yemen, Senegal and Sudan. He has also worked with the World Bank, in Washington DC on investment appraisals for major development projects.

Robert is a Master of Business Administration (Henley Management College), a Member of the Chartered Institute of Marketing, Chartered Engineer and a Member of the Institution of Civil Engineers. In 2010, Robert received a Distinguished Service Certificate from the BSI for services to national and international project management standards, and in 2013 he was made an Honorary Fellow of the Association for Project Management.