

Introduction

This chapter covers:

- what PRINCE2 Agile is for
- PRINCE2 Agile is for projects only
- projects and business as usual

1 Introduction

1.1 What is PRINCE2 Agile?

PRINCE2 Agile describes how to configure and adapt PRINCE2 so that PRINCE2 can be used in the most effective way when combining it with **agile behaviours**, concepts, frameworks and techniques.

1.2 PRINCE2 Agile is for projects only

PRINCE2 and PRINCE2 Agile are only suitable for use on **projects**, whereas agile can be used for projects and routine ongoing work as well. Throughout this manual, routine ongoing work is referred to as ‘business as usual’ (BAU) and covers areas such as ongoing product development, product maintenance and continual improvement.

The distinction between project work and BAU work (see Table 1.1 and Figure 1.1) is important because some of the agile ways of working need to be applied differently in each situation. Therefore, when carrying out a piece of work, it is important to understand the type of work being undertaken, to ensure that it is addressed in the appropriate way and that agile is used appropriately.

Table 1.1 The different characteristics of a project and BAU work

Project characteristics	BAU characteristics
Temporary	Ongoing
Team is created	Stable team
Difficult	Routine
A degree of uncertainty	A degree of certainty



Tip

AXELOS's *Managing Successful Programmes* (MSP) provides best-practice guidance for managing related projects and activities in programmes of work that deliver business benefits through new capabilities.

1.2.1 What does BAU look like?

BAU work would typically be repeatable routine tasks that can be carried out by people with the appropriate technical skills without needing to be managed by a **project manager**. An example of this would be when modifications or enhancements need to be made to an existing product and the timescales are relatively short. There would usually be a long list of these tasks arriving regularly throughout the lifespan of the product. There may be an established team dedicated to this work.

1.2.2 What does a project look like?

A project is a temporary situation where a team is assembled to address a specific problem, opportunity or change that is sufficiently difficult that it cannot be handled as BAU. It may even be a collection of BAU items handled collectively. An example of a project would be the creation of a new product or service; there may be a need to engage many stakeholders and a significant amount of uncertainty exists. The project team may be based in different locations, the team personnel may change, the project may last a long time and it may be part of a wider **programme** of work. Importantly, it needs to be managed by a project manager.

Figure 1.1 illustrates the different characteristics of project work in comparison with BAU work. A project has defined **stages** for upfront work before any delivery activity commences. It also has layers of **project management** and project direction to ensure the correct output is ultimately reached. By the end of a project, at which point the project team disbands (or moves to other work), the product created will have gone into operational use. From then on it may be maintained and enhanced in a BAU environment.

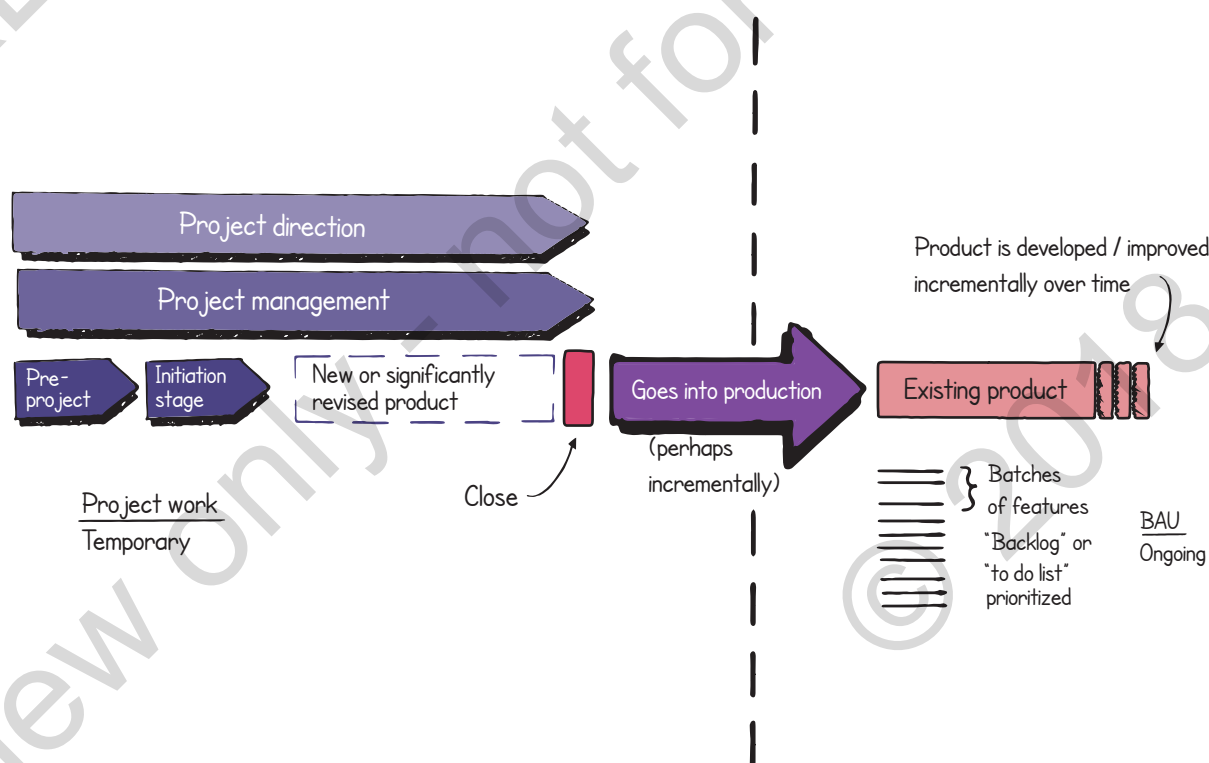


Figure 1.1 The difference between project work and BAU work

Note: PRINCE2 Agile can be used on the left-hand side of the dashed line only (i.e. for projects). Agile can be used on both sides (i.e. used on projects and for BAU).



Definition: Timebox

A finite period of time when work is carried out to achieve a goal or meet an objective. The deadline should not be moved, as the method of managing a timebox is to prioritize the work inside it. At a low level a timebox will last a matter of days or weeks (e.g. a sprint). Higher-level timeboxes act as aggregated timeboxes and contain lower-level timeboxes (e.g. stages).

In a BAU environment, the list of work is prioritized in some form and may be batched into **timeboxes**. As the work is completed the existing product evolves, continually, over time.

Although PRINCE2 Agile is only suitable for projects, it uses a wide range of agile behaviours, concepts, frameworks and techniques that are also used in a BAU environment.

2

An overview of agile

This chapter covers:

- some history behind agile and what it is today
- agile basics including the frameworks, behaviours, concepts and techniques

2 An overview of agile

2.1 Introduction

The term 'agile' is very broad and is viewed in many different ways throughout the agile community. There is a set of well-known frameworks referred to as 'agile methods' and there are also well-known behaviours, concepts and techniques that are recognized as characterizing the agile way of working. But there is no single definition of agile that accurately encapsulates them all, although the Agile Manifesto (see Figure 2.1) comes the closest to achieving this.

2.1.1 Some history

The term 'agile' was created in 2001 (www.agilemanifesto.org) when a group of 'independent thinkers around software development' came together to talk about an alternative to the heavyweight, document-driven processes that existed at the time. Known as the '**waterfall method**' (see Figure 2.2), these old-fashioned processes comprised a sequence of technical phases that were slow and struggled to respond to changing **requirements**, particularly when they were mired in too much detail from the start.

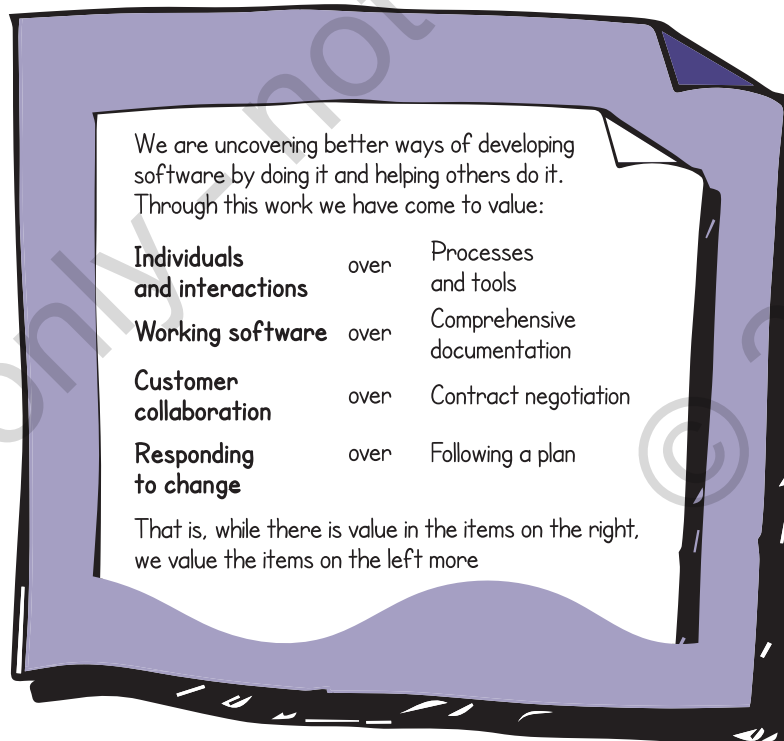


Figure 2.1 The Agile Manifesto

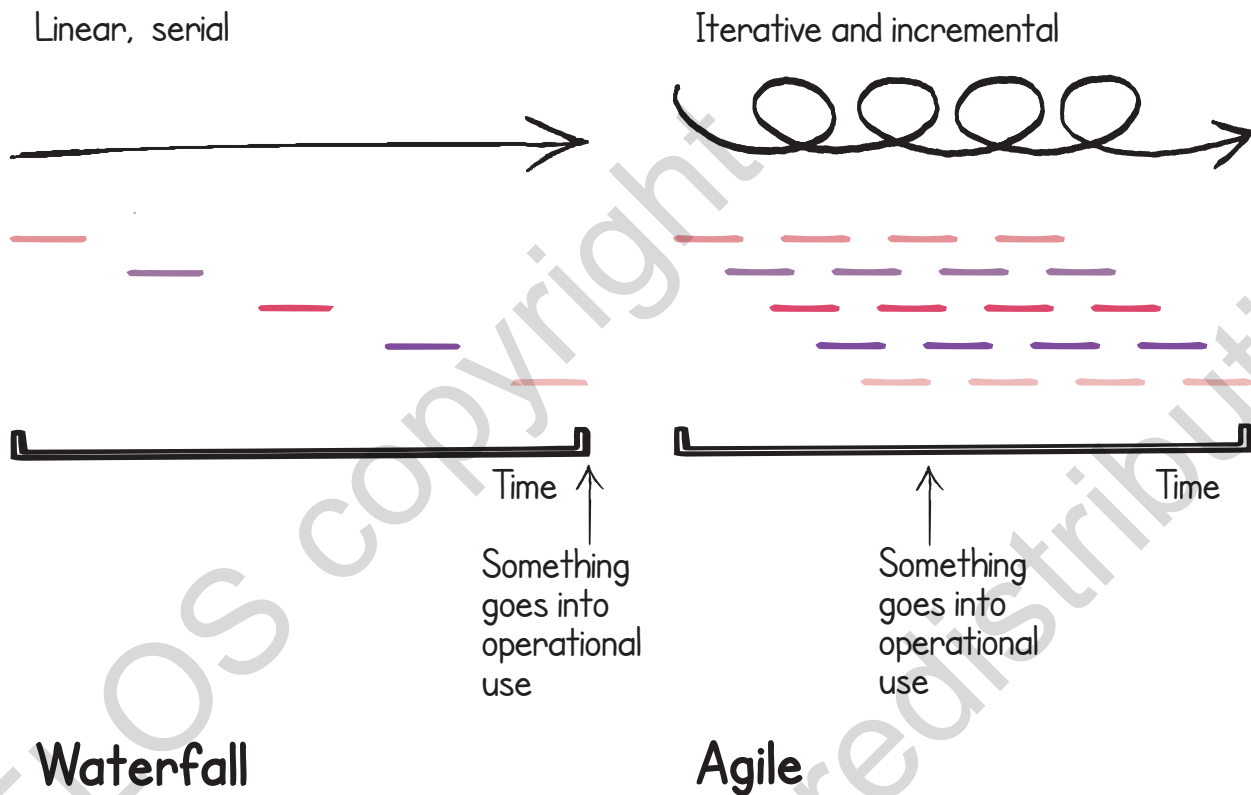


Figure 2.2 The contrast between waterfall and agile phases

The group was already working in ways that were eventually described as agile. An output from this meeting was the Manifesto for Agile Software Development, or the 'Agile Manifesto' as it is more commonly known, and its impact and success have been quite dramatic. The Agile Manifesto is summarized in Figure 2.1; it also contains 12 principles which are listed in Appendix E.1. It is important to appreciate the intent of the final two lines of the Agile Manifesto: it is a case of relative importance of the values, and not a case of 'good' or 'bad'.

The reason why agile became so popular was that it helped to address the new demands being placed on how software was delivered. Software needed to be produced more frequently while at the same time being of the appropriate **level of quality** to meet the demands of new technologies, the internet and the digital era. In contrast to the waterfall way of working, agile phases are smaller, more iterative and incremental (see Figure 2.2).

By definition, the Agile Manifesto only applies to developing software; most of its underlying principles appear to suggest that this is in the context of the continual timeboxed development of a software product. Although it was created as a way to develop software, it has since been recognized as a successful approach beyond software development. Many people use the Agile Manifesto, replacing the word 'software' with 'products' or 'solutions'.

2.1.2 Agile today

Agile has come a long way since 2001 and is no longer just 'an IT thing'. It now includes situations that are large scale, complex in nature and happening in a wide array of contexts far beyond software development.

Nowadays, most organizations are aware of the term agile, and every organization should have a strategy in place to adopt it to some degree. For many years it was seen as a niche area; it is now mainstream and is used by organizations that are large and small, old and new, public sector and private sector.

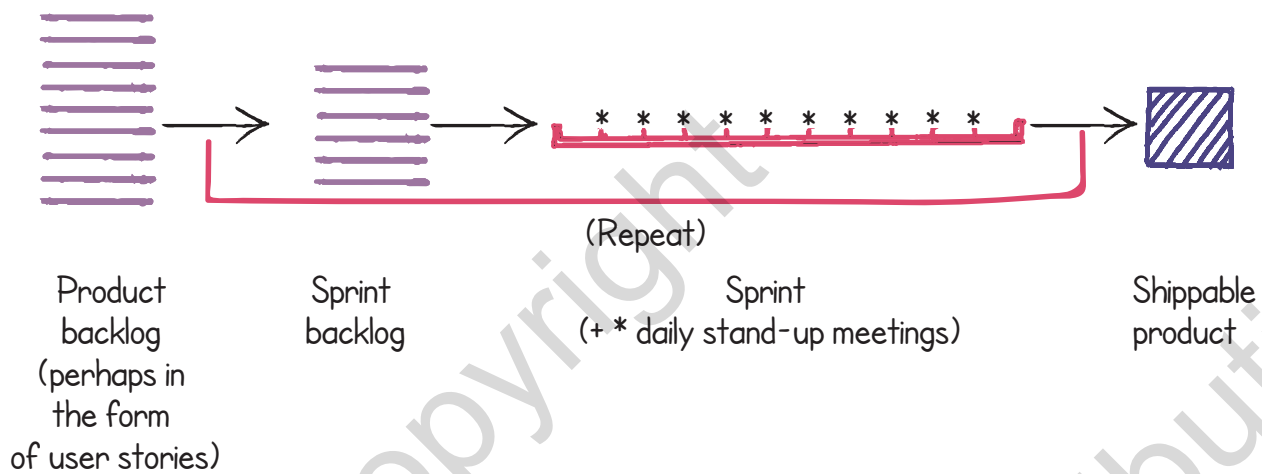


Figure 2.3 A basic 'backlog' and 'sprint' structure for delivering software

2.2 Agile basics

When combining PRINCE2 with agile it is important to know what agile is. An inconsistent view of the basics of agile will make it difficult to combine the two; for example, if someone in an organization thinks that agile can only be used on the IT part of a project, whereas someone else thinks it can be applied across the whole project, then this will present a problem.

A basic view of agile could generally be seen as using one or more of the following:

- a timeboxed and iterative approach to delivering software
- a collection of techniques such as daily **stand-up meetings**, **sprints** and user stories
- the Scrum framework (see Table 2.1).

Figure 2.3 shows a very common structure that is used when working in an agile way for developing software. In simple terms, new **features** for a product are held in a prioritized list called the product **backlog**. The list may be made up of user stories, which are structured in a way that describes who wants the feature and why. The team that will build the features decides which items from the top of the product backlog they can create in a timeframe of typically 2–4 weeks (which is known as a sprint). The work that the team think they can achieve during the sprint is held in a list called a sprint backlog. Each day throughout the sprint, a meeting is held to assess progress. At the end of a sprint new features should have been created and they may go into operational use.



Definition: Release

The set of products in a handover. The contents of a release are managed, tested and deployed as a single entity.

In PRINCE2 Agile, a release is typically a container for more than one low-level timebox (e.g. a sprint). This is not always the case as the act of releasing features into operational use may happen more regularly (e.g. after each sprint or several times during a sprint). The term 'deployment' is sometimes used in agile and has a similar meaning, although it is not used in PRINCE2 Agile.

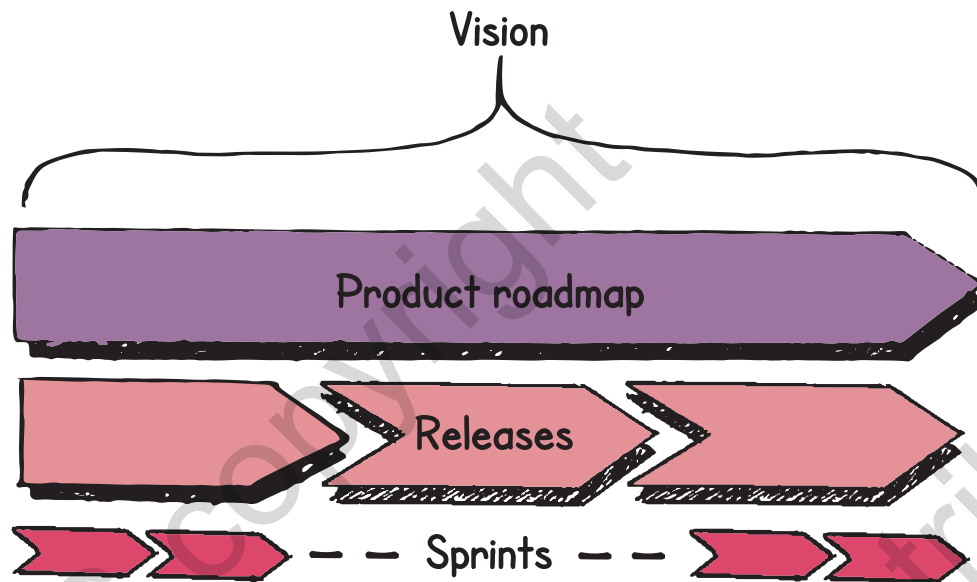


Figure 2.4 Sprints may exist within a wider context

The output (i.e. the new features) is reviewed along with the way the team worked to achieve that output. This basic structure may exist within an overall approach that includes a **vision**, a **product roadmap** (which is a plan of how a product will evolve) and a series of **releases** (see Figure 2.4).

The two examples represented in Figures 2.3 and 2.4 provide a typical view of agile, although it is somewhat limited. A more comprehensive view would include:

- IT and non-IT situations
- large and small projects as well as routine 'business as usual' (BAU) tasks
- **flow-based** working as well as timeboxing.

Further to this there also needs to be a wider mindset and a collection of behaviours that enable the agile way of working to thrive.



Definition: Flow-based

This avoids the use of partitioning work into timeboxes and manages work by using a queue. Work is then continually pulled into the system (which may itself be a high-level timebox) and moves through various work states until it is done.

2.2.1 Agile frameworks

There is a family of frameworks (also referred to as methods or approaches) that are generally recognized as being agile. However, some are only applicable to IT situations. A summary of the most well known is shown in Table 2.1.

Table 2.1 The most well-known agile methods and approaches

Term	Brief description
ASD (Adaptive Software Development)	(IT only) Iterative development process (Highsmith, 2000).
Crystal	(IT only) Iterative development method (Cockburn, 2001).
DAD (Disciplined Agile Delivery)	(IT only) An enterprise-wide scalable process framework described as 'a process decision framework that is a people-first, learning-oriented hybrid agile approach to IT solution delivery', that has 'a risk-value delivery lifecycle, is goal-driven, is enterprise aware and is scalable.' See http://www.disciplinedagiledelivery.com
DevOps	(IT only) A collaborative approach between development and operations aimed at creating a product or service where the two types of work and even the teams merge as much as possible.
DSDM (Dynamic Systems Development Method)/ AgilePM	An agile project framework that focuses on the iterative delivery of business systems through the use of timeboxing and continual business involvement. It has a defined process and corresponding set of products, a set of roles that operate at all levels of a project, eight guiding principles and a collection of key techniques that can be used throughout a project.
FDD (feature-driven development)	(IT only) Iterative software development process focusing on features.
Kanban	A way to improve flow and provoke system improvement through visualization and controlling work in progress .
Lean	An approach that focuses on improving processes through maximizing value by eliminating waste (such as wasted time and wasted effort).
Lean Startup	Originally an approach to creating and managing start-up companies, but now applied to any business, to help them deliver products to customers quickly.
SAFe (Scaled Agile Framework)	(IT only) Large-scale application of agile across an organization. PRINCE2 and PRINCE2 Agile could be used in SAFe where a piece of work is of a sufficient size or level of difficulty that it should be run as a project.
Scrum	An iterative timeboxed approach to product delivery that is described as 'a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value' (see Appendix H).
XP (eXtreme Programming)	(IT only) Iterative software engineering practice that can be used on its own but often exists in tandem with Scrum or Kanban, where XP covers the creation of the software and Scrum or Kanban is used as an overarching framework to control the work.



Tip

PRINCE2 Agile regards agile as a family of behaviours, concepts, frameworks and techniques.

2.2.2 Agile behaviours, concepts and techniques

Along with the agile frameworks there are a variety of behaviours, concepts and techniques that are seen as being part of the agile way of working. Examples are shown in Table 2.2 (some of which are defined in the glossary) but the table only provides a few illustrative examples of what is seen as agile. It is not a complete list and it is not necessary to be strict on the exact terms used (e.g. whether or not something is a technique or a behaviour).

Table 2.2 Typical agile behaviours, concepts and techniques

Term	Examples	Similar terms
Behaviours	Being collaborative, self-organizing, customer-focused, empowered, trusting not blaming.	Principles, values, mindset
Concepts	Prioritizing what is delivered, working iteratively and incrementally, not delivering everything, time-focused, 'inspect and adapt'. Kaizen . Limiting work in progress (WIP).	Fundamentals
Techniques	Burn charts, user stories, retrospectives , timeboxing, measuring flow.	Practices, tools

PRINCE2 and PRINCE2 Agile do not favour one agile approach over any other; this is sometimes referred to as being 'agile agnostic'. With due care and consideration, they can engage with agile in all of its many forms to provide a holistic project management approach that can be tailored to suit a wide variety of conditions and working environments.

Acknowledgements and further research

Cockburn, A. (2001) *Agile Software Development: Software Through People*. Addison Wesley.

Highsmith, J.A. (2000) *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House, New York.

3

The rationale for blending PRINCE2 and agile

This chapter covers:

- who PRINCE2 Agile is for and who will benefit from it
- when and where to apply PRINCE2 Agile
- what PRINCE2 Agile consists of

3 The rationale for blending PRINCE2 and agile

PRINCE2 is the most commonly used project management approach in the world, and it is increasingly being used in conjunction with agile. As more organizations adopt agile, the need for specific guidance on how to use PRINCE2 in an agile context has grown accordingly.

In simple terms, PRINCE2 and agile each have their own strengths and when combined they complement each other and create a holistic approach to managing projects in an agile way.

The strength of PRINCE2 lies in the areas of project direction and project management. However, it provides little focus on the field of product delivery.

Conversely, agile has a very strong focus on product delivery but relatively little on project direction and project management (see Figure 3.1).

Therefore, when PRINCE2 and agile are combined, all three areas in Figure 3.1 are addressed.

It is essential to see this combination as a blend and a mixture as opposed to PRINCE2 and agile working in parallel. Those directing and managing a project in an agile context need to adopt agile disciplines and behaviours. Equally, those using agile to deliver need to integrate seamlessly with the PRINCE2 ethos of staying in control by empowering people and ensuring that the project remains viable.

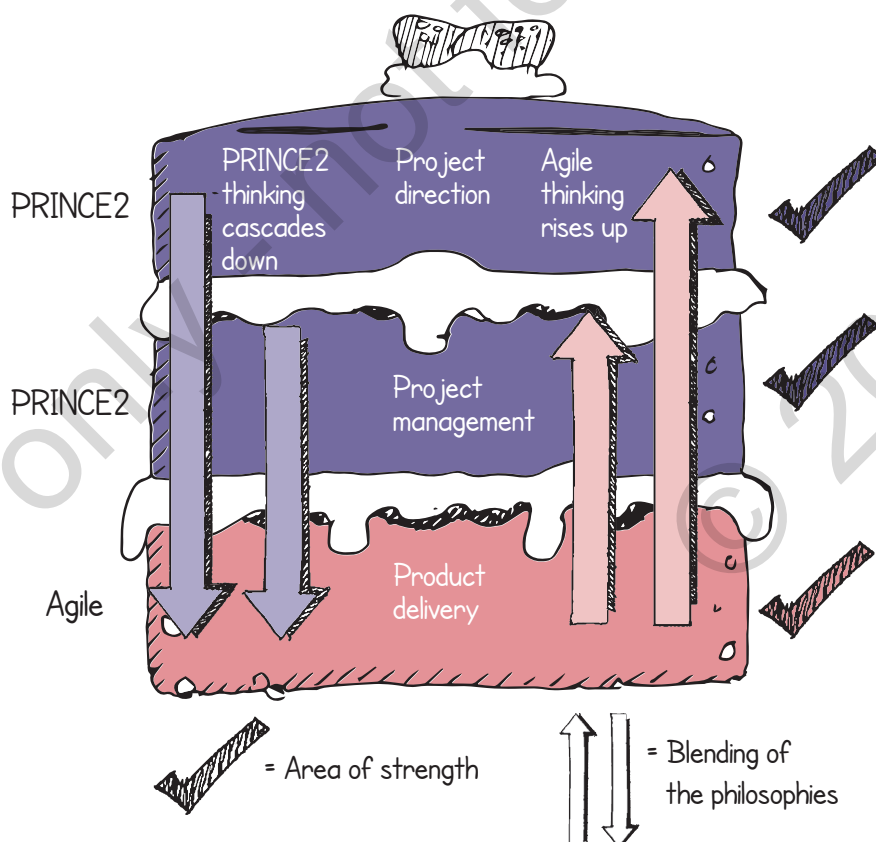


Figure 3.1 Blending PRINCE2 and agile together

Historically, the majority of the applications of agile related to ongoing product development, and this would typically be in an IT context. However, the use of agile has now gone beyond IT and is being used in more challenging situations. Therefore, as the need for agile to work in a project context increases, the established and internationally recognized methods for project management have needed to provide guidance on how they work with agile.

PRINCE2 Agile provides guidance for tailoring PRINCE2 to work in the most effective way in an agile context.

3.1 Who is PRINCE2 Agile for?

The primary purpose of this guidance is to help and support the existing PRINCE2 community and in particular PRINCE2 practitioners. Anyone directing or managing a PRINCE2 project who is using agile will benefit from this guidance. Anyone involved with, or impacted by, a PRINCE2 project that is using agile will also benefit from this guidance (e.g. **project support, quality assurance**).

Organizations and individuals outside the PRINCE2 community can also benefit from this guidance in one of the following ways:

- Those who currently have experience with agile may want to become familiar with, and perhaps ultimately adopt, an internationally recognized standard for project management along with their existing agile capability.
- Those who are relatively new to agile may want to become familiar with, and perhaps ultimately adopt, an internationally recognized standard for project management as they evolve their agile capability.

PRINCE2 Agile is not a substitute for PRINCE2. However, for those outside the PRINCE2 community, *PRINCE2 Agile* may provide additional guidance to help them adopt PRINCE2 for all of their projects, irrespective of whether or not they involve agile.

3.2 Which communities will benefit from PRINCE2 Agile?

PRINCE2 Agile is aimed at the communities shown in Table 3.1.

Table 3.1 The communities on which PRINCE2 Agile is focused

Focus of PRINCE2 Agile	Community	Description	Possible goals
Primary	PRINCE2 wanting to adopt (more) agile	PRINCE2 organizations, departments or individuals who wish to adopt agile.	They wish to integrate agile into their existing PRINCE2 working practices in order to benefit from this way of working.
Primary	PRINCE2 encountering agile	PRINCE2 organizations, departments or individuals who are encountering other organizations, departments or individuals using agile in some form.	They wish to understand what agile is and how it works so that they can tailor PRINCE2 appropriately and integrate with it.
Primary	PRINCE2 practising agile	PRINCE2 organizations, departments or individuals who are practising agile in some form.	They wish to adopt a formal standard for using agile in a PRINCE2 context.
Secondary	Agile wanting to adopt PRINCE2	Organizations, departments or individuals who have some agile experience and wish to adopt PRINCE2.	They wish to adopt an industry standard approach to project management and project governance to complement their existing agile working practices.
Possible area of focus	Mature agile	Organizations, departments or individuals who have adopted agile and have reached high levels of maturity for both BAU and project work.	They wish to become familiar with PRINCE2 Agile in order to add to their existing body of knowledge.

3.3 When and where can I apply PRINCE2 Agile?

PRINCE2 Agile provides governance and project management controls that are suitable for some situations but not others as shown in Table 3.2.

Table 3.2 The situations for which PRINCE2 Agile is suitable

Situation	Description	Suitability of PRINCE2 Agile
Agile is not used for projects, or is at a basic level	Either agile is not used for projects, or an agile way of working exists and is evolving with a limited level of maturity in terms of processes and behaviours (e.g. how formalized and documented they are and how predominant they are).	Suitable for organizations, departments and the individuals working within them.
Agile is used for BAU	An agile way of working exists for use with ongoing routine development of an existing product. Most, if not all, work is handled this way.	Not suitable for organizations and departments. However, it is suitable to support and enhance this style of working (i.e. product delivery) in a project context, as some work may be better suited to being managed as a project. Additionally, it may help individuals' career progression.
A mature level of agile is used for projects and BAU	An agile way of working exists that is taking place in a mature agile environment where processes are formalized and repeatable and agile behaviours are predominant.	Suitable for projects, but may be of limited use. Could be of use to support information to enhance an existing body of knowledge. Organizations, departments and individuals who are conversant with PRINCE2 Agile will be suitable for this situation as they will understand it quickly and integrate with it easily, albeit using different terminology.

3.4 Course of action for each community in each situation

Table 3.3 outlines the most appropriate course of action that the communities listed in Table 3.1 would need to take in terms of using PRINCE2 Agile, when faced with the situations described in Table 3.2.

Table 3.3 The relationship between each community and each situation with respect to PRINCE2 Agile

Community	Situation		
	Agile is not used for projects, or is at a basic level	Agile is used for BAU	A mature level of agile is used for projects and BAU
PRINCE2 wanting to go agile, or encountering, or practising agile	Use PRINCE2 Agile.	Should some of the BAU work be handled as a project? If so, use PRINCE2 Agile for this.	Potentially, PRINCE2 Agile may be of use, although the existing agile process may suffice. See note 1.
Agile wanting to adopt PRINCE2	Use PRINCE2 Agile.	Should some of the BAU work be handled as a project? If so, use PRINCE2 Agile for this.	Potentially PRINCE2 Agile may be of use although the existing agile process may suffice. See note 2.

Notes:

- Mapping of similar terms may be of benefit (e.g. product board may be an equivalent to project sponsor).
- A situation where a mature level of agile is used for projects may potentially benefit from using PRINCE2 Agile, unless it is felt that the existing governance and project management controls are sufficient. Using PRINCE2 Agile may be appropriate if a more challenging piece of work is being undertaken or a project involves a customer, supplier or another part of the organization that is using PRINCE2 or PRINCE2 Agile. In any of these cases PRINCE2 Agile could be adopted, or similar terms could be mapped between the different approaches.

3.5 What does PRINCE2 Agile consist of?

PRINCE2 comprises an integrated set of principles, themes and processes that are tailored to the specific needs of a project (see Figure 3.2).

PRINCE2 Agile provides guidance on tailoring PRINCE2 in an agile context and covers:

- how to apply the principles and tailor the themes and processes
- how to produce the PRINCE2 management products
- how to map the common agile roles to the PRINCE2 project management team structure
- how to incorporate the fundamental agile behaviours, concepts and techniques into PRINCE2
- what areas are of particular significance when using agile and need specific focus.

This applies to all levels of a PRINCE2 project (i.e. project direction, project management and product delivery).

Agile encapsulates a wide collection of frameworks, and PRINCE2 Agile incorporates and references several of them. A few of these methods and approaches receive particular attention in PRINCE2 Agile because they are seen as very popular or good practice. They are Scrum, Kanban and Lean Startup (see Table 2.1 for a brief description).

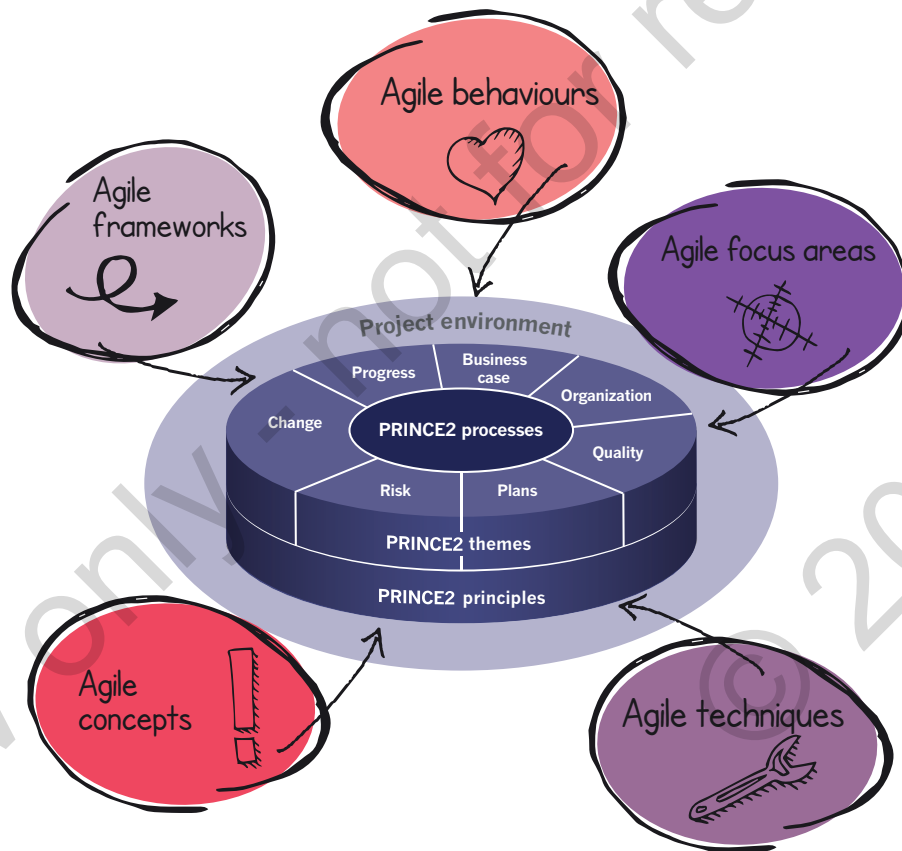


Figure 3.2 Tailoring PRINCE2 by blending in the agile ingredients

3.6 Important points about PRINCE2 Agile and this manual

PRINCE2 has been used and implemented in a wide variety of situations and has evolved significantly since it was first published in 1996. Agile has had a similar journey, but also comes in many different forms.

In order to correctly understand PRINCE2 Agile and this manual, the reader must be aware of the following eight guidance points which are intended to provide clarity where there is potential for ambiguity, and accuracy where there may be misconceptions. These points are summarized in Table 3.4 for quick reference.

1. All references to PRINCE2 refer to the edition created in 2017 (*Managing Successful Projects with PRINCE2*). This version is already enabled to work with agile; it just needs to be tailored to suit any given project. Configuring PRINCE2 in the most effective way for agile is a matter of where to put emphasis and where to add further levels of detailed guidance. Nothing needs to be removed or significantly changed, as this is achieved through blending agile into PRINCE2 and tuning PRINCE2 appropriately.
2. PRINCE2 allows for any style of working, including environments that involve high levels of informality, collaboration and trust. It should be understood that PRINCE2 should not be thought of as a 'traditional project management approach' in the stereotypical sense of being predominantly 'waterfall', 'big design up front', 'bureaucratic' and using a 'command and control' culture. PRINCE2 does not suggest that a project should be run in this way, and much of its guidance is to the contrary.
3. Most of the heritage and thinking behind agile has come from IT and software development, but PRINCE2 Agile does not assume an IT context; although it can be used in an IT context, it is not an IT framework or an IT method.
4. Many agile approaches and frameworks are created solely for IT situations; for example, eXtreme Programming (XP) and the Scaled Agile Framework (SAFe). PRINCE2 Agile will only make passing reference to IT-only frameworks (see Table 2.1 for brief descriptions).
5. The most well-known agile framework is Scrum, and PRINCE2 Agile is written with the view that although Scrum can rightly be described as being 'agile', the converse is not the case: it is not true to describe agile as 'using the Scrum framework'. Other frameworks exist and a framework is only part of the agile way of working.
6. Although agile appears in many forms, the use of the Scrum framework and, to a lesser extent, the Kanban framework (either separately or in combination) make up the vast majority of what agile practitioners use. There are many other frameworks and approaches but when this manual uses expressions such as 'commonly used in agile' or 'widely used in agile' it will usually be referring to either of these two frameworks. It is important to note that Scrum and Kanban are not project management frameworks, and a project manager role is not defined in either. On their own, and in isolation, they cannot be used to manage a project. They can, however, be used on a project as part of an approach to delivering products, as long as they are contained within a wider project management framework such as PRINCE2.
7. The term 'agile' when used on its own in this manual refers to a general family of behaviours, concepts, frameworks and techniques that is widely accepted throughout the agile community as being part of the agile way of working. The terms 'behaviours, concepts, frameworks and techniques' also encapsulate other similar terms such as methods, principles, values, mindsets and approaches.

8. PRINCE2 Agile does not see working in an agile way as a binary condition (i.e. you either are or you are not working in an agile way). It always sees agile as a question of how much (or how little) it can be used according to the situation that exists. To illustrate this point, *PRINCE2 Agile* does not refer to 'agile projects' as this would infer that some projects are agile, whereas others are not. *PRINCE2 Agile* is written with the view that agile behaviours, concepts, frameworks and techniques can be applied to any project.

Table 3.4 Summary of the key points in Chapter 3

Key point	
1	PRINCE2 (2009 edition onwards) is already enabled for use with agile.
2	PRINCE2 is suitable for any style of project and is not a 'traditional' project management approach as is typically contrasted to agile.
3	PRINCE2 Agile is for any project and not just for IT projects.
4	'IT-only' frameworks and techniques are mentioned in PRINCE2 Agile but not extensively.
5	There is much more to agile than the Scrum framework. Agile is not Scrum.
6	The most 'commonly used' agile approaches are Scrum and Kanban, but they are not suitable for managing a project in isolation. However, they can be effectively used in a project context.
7	The term 'agile' (in this manual) refers to a family of behaviours, concepts, frameworks and techniques.
8	Using agile on a project is not a question of 'yes or no': it is about 'how much'.

3.7 Beware of prejudice!

It would be understandable to think that bringing more control and governance into the agile domain could prove counter-productive. However, PRINCE2 Agile represents a marriage that is based on the opposite view: that control and governance allow agile to be used in more situations such as those involving multiple teams or complex environments.

A fighter aircraft is built with a deliberately unstable airframe. This instability gives it agility and allows it to change direction easily and adapt quickly to situations. However, to do this still requires control and governance!

This personifies PRINCE2 Agile.



Image 3.1 An agile fighter aircraft