PRINCE2 Agile® Foundation and Practitioner

PRINCE2 and Agile Concepts

By the end of this lesson, you will be able to:

- Distinguish between projects and business as usual (BAU)
- Describe agile and its common approaches
- Explain the Kanban method and its general practices
- Identify the core concepts of Lean Startup
- Define workshop

PRINCE2 Agile: Only for Projects

PRINCE2, PRINCE2 AGILE, AND AGILE

PRINCE2 and PRINCE2 Agile

Are only suitable for use on projects

Agile

Can be used for both projects and routine ongoing work

Routine ongoing work is referred to as *BAU*. It includes ongoing product development, product maintenance, and continual improvement.

PROJECT VS. BAU

It is important to distinguish between project work and BAU work as some of the agile ways of working need to be applied differently in each situation.

The table illustrates the different characteristics of a project and BAU work.

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

Table 1.1 The different characteristics of a project and BAU work

BUSINESS AS USUAL (BAU)

- BAU work will typically be repeatable routine tasks.
- People with appropriate technical skills can carry out BAU tasks.
- It does not require a project manager.

Example: When modifications or enhancements need to be made to an existing product and the timescales are relatively short

PROJECT

It is a temporary situation where a team is assembled to address a specific problem, opportunity, or change that is sufficiently difficult to be handled as BAU.

It may even be a group of BAU items handled collectively.

PROJECT WORK AND BAU WORK

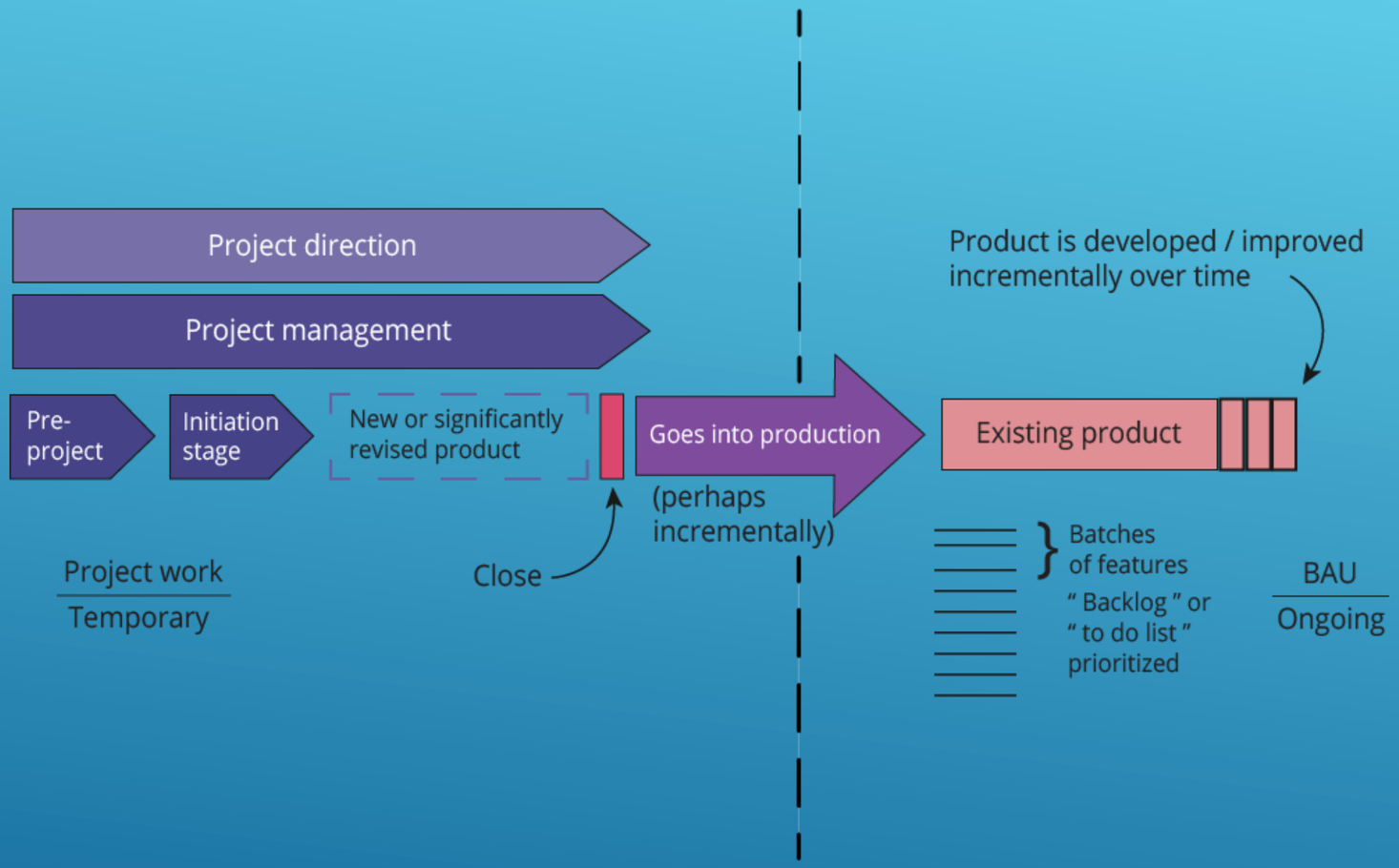


Figure 1.1 The difference between project work and BAU work

PROJECT WORK AND BAU WORK

- In a BAU environment, prioritize the list of work and batch it into timeboxes
- Although PRINCE2 Agile is only suitable for projects, it uses a wide range of agile behaviors, concepts, frameworks, and techniques that are also used in a BAU environment

Timebox

It is 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.

Agile: Overview

AGILE: OVERVIEW

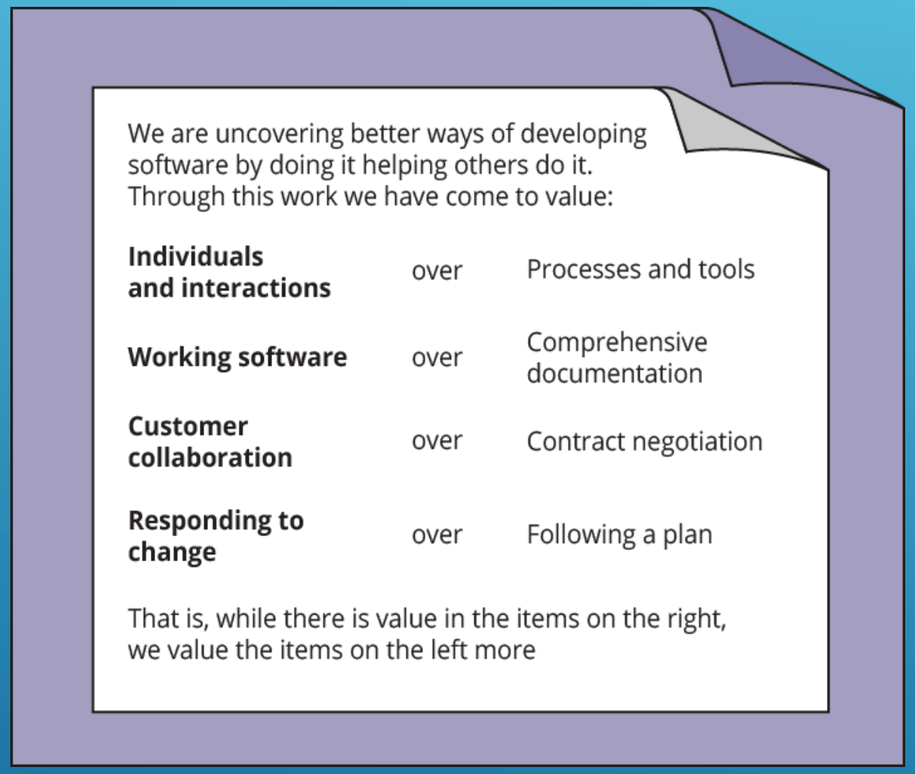
The term *agile* is very broad and is viewed in many different ways throughout the agile community.

There is a set of well-known frameworks referred to as *agile methods.*

There are also well-known behaviors, concepts, and techniques that are recognized as characterizing the agile way of working.

AGILE: OVERVIEW

The *Agile Manifesto* comes the closest to the definition of agile that accurately encapsulates agile methods.



We are uncovering better ways of developing software by doing it helping others do it. Through this work we have come to value:

Individuals and interactions	over	Processes and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

That is, while there is value in the items on the right, we value the items on the left more

Figure 2.1 The Agile Manifesto

AGILE: HISTORY

The term agile was created in 2001 by a group of independent thinkers around software development as an alternative to waterfall method.

AGILE: HISTORY

Agile became popular as it helped address the new demands being placed on how software was delivered.

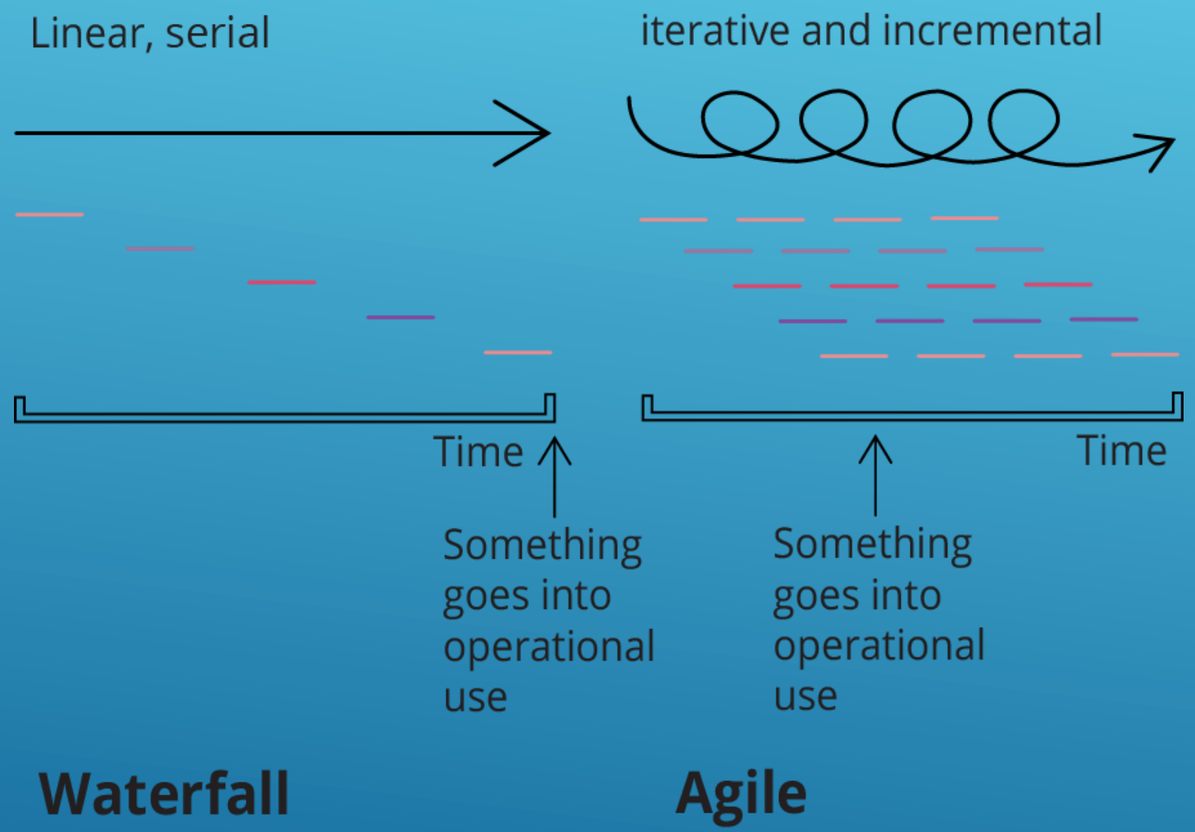


Figure 2.2 The contrast between waterfall and agile phases

AGILE: HISTORY

The group was already working on ways that were eventually described as agile. An output from this meeting was the Manifesto for Agile Software Development, or the *Agile Manifesto.*

The Agile Manifesto

THE AGILE MANIFESTO

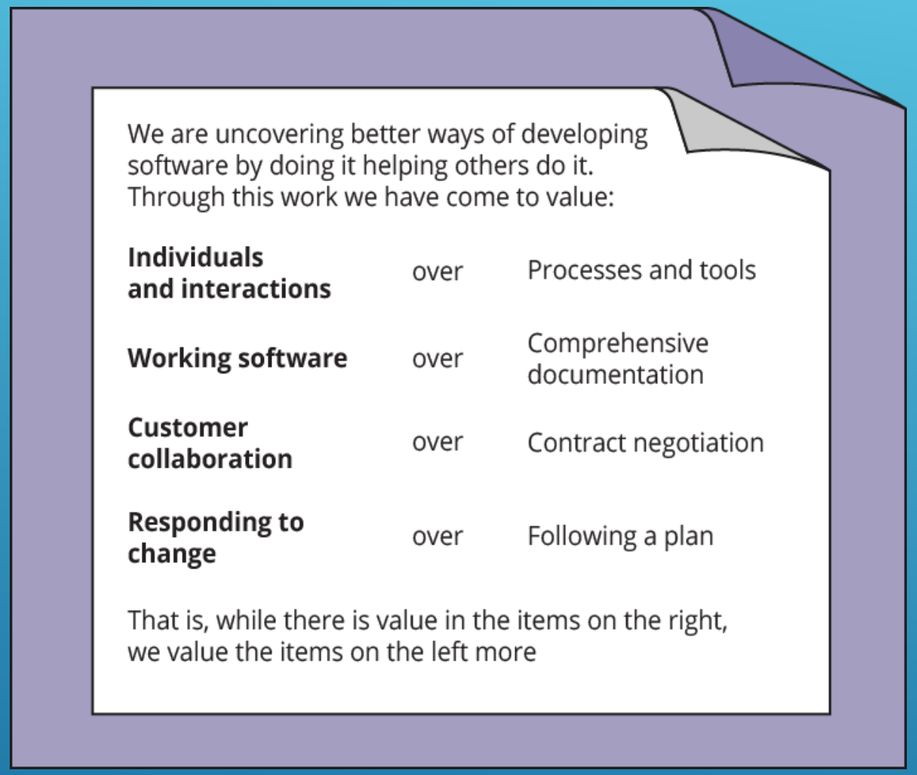
We are uncovering better ways of developing software by doing it helping others do it. Through this work we have come to value:

Individuals and interactions	over	Processes and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

That is, while there is value in the items on the right, we value the items on the left more

Figure 2.1 The Agile Manifesto

THE AGILE MANIFESTO



We are uncovering better ways of developing software by doing it helping others do it. Through this work we have come to value:

Individuals and interactions	over	Processes and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

That is, while there is value in the items on the right, we value the items on the left more

Figure 2.1 The Agile Manifesto

- Agile Manifesto contains 12 principles.
- 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.*

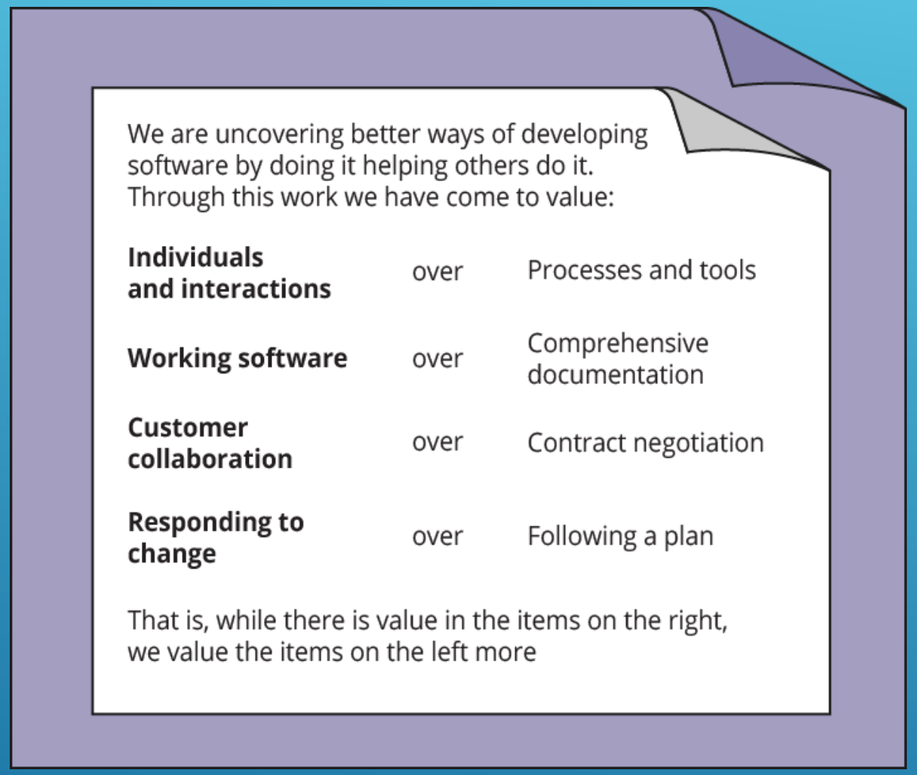
THE AGILE MANIFESTO: 12 PRINCIPLES

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

THE AGILE MANIFESTO: 12 PRINCIPLES

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

THE AGILE MANIFESTO



We are uncovering better ways of developing software by doing it helping others do it. Through this work we have come to value:

Individuals and interactions	over	Processes and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

That is, while there is value in the items on the right, we value the items on the left more

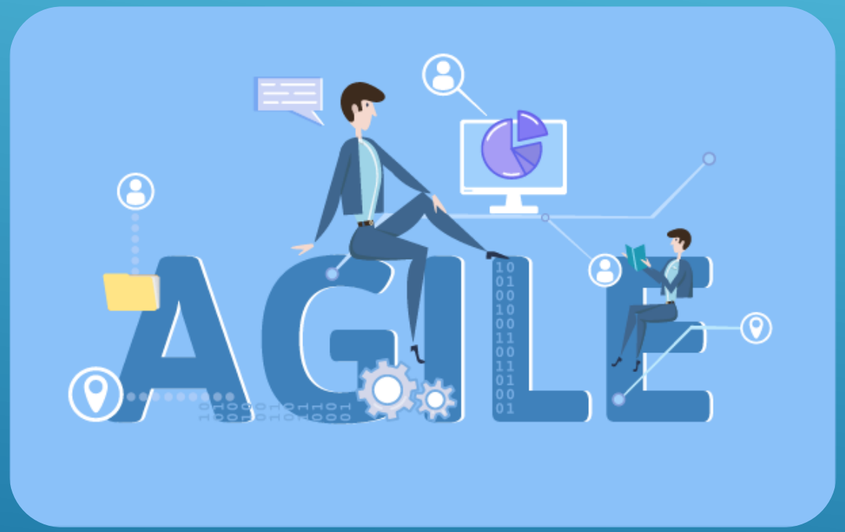
Figure 2.1 The Agile Manifesto

- By definition, it only applies to developing software.
- Most of its underlying principles suggest that it is in the context of the continual timeboxed development of a software product.
- Many people use the Agile Manifesto, replacing the word *software* with *products* or *solutions*.

Agile Basics

AGILE TODAY

Agile is no longer just *an IT thing*. It includes situations that are large scale, complex in nature, and happening in a wide array of contexts beyond software development.



Agile is now used by organizations that are:

- Large and small
- Old and new
- In the public sector
- In the private sector

Most organizations these days are aware of the term agile and every organization should have a strategy in place to adopt it to some extent.

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.

Agile can generally be seen as using:

- 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

AGILE BASICS

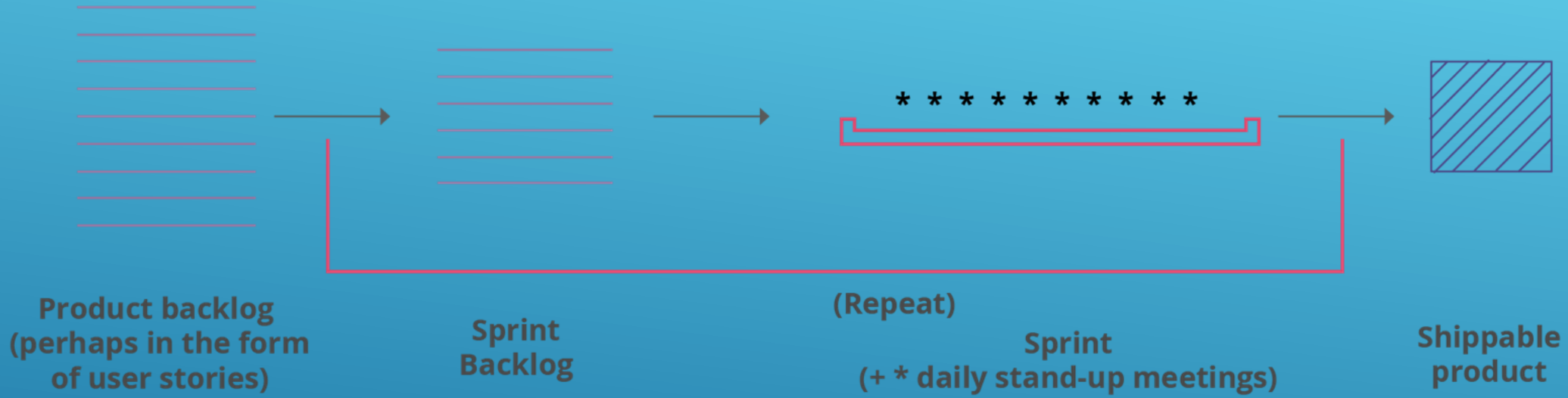


Figure 2.3 A basic "backlog" and "sprint" structure for delivering software

The new features are reviewed along with the way the team worked to achieve that output

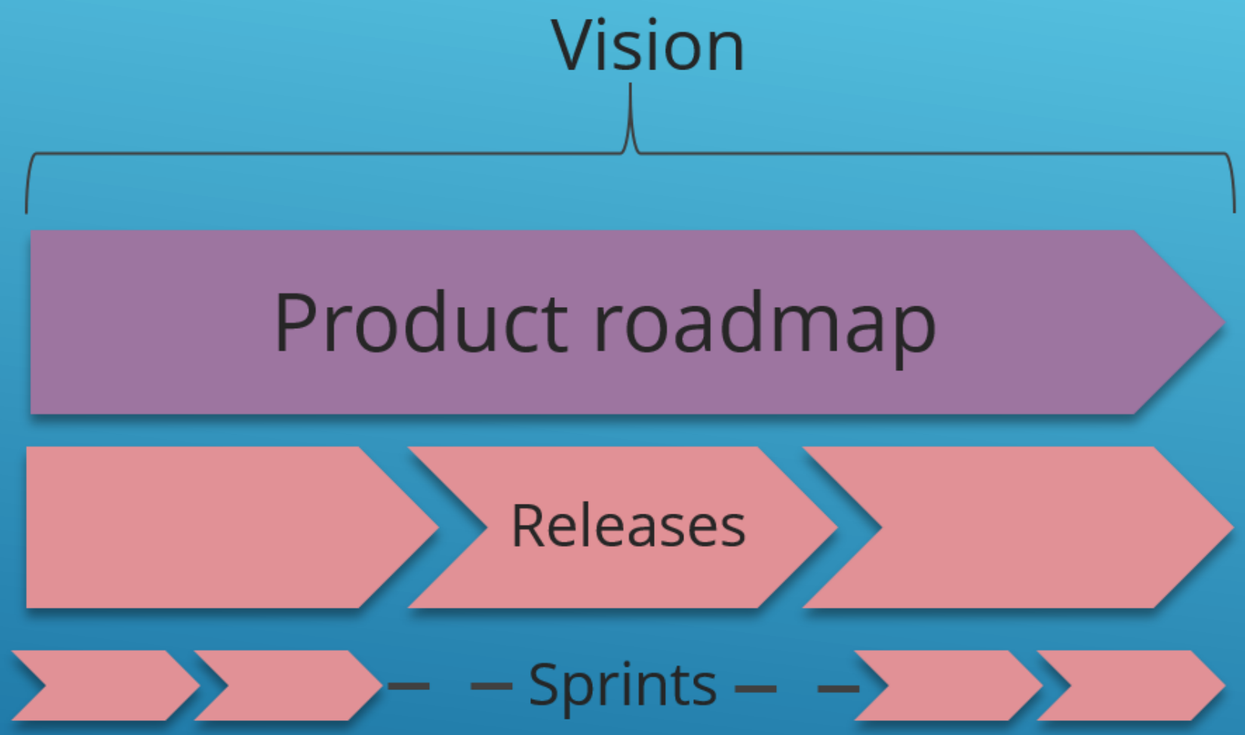


Figure 2.4 Sprints may exist within a wider context

AGILE BASICS

Release

It is 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, for example, a sprint.

This is not always the case as the act of releasing features into operational use may happen more regularly, for example, 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.

AGILE BASICS

A more comprehensive view of agile would include:

- IT and non-IT situations
- Large and small projects as well as routine BAU tasks
- Flow-based working as well as timeboxing

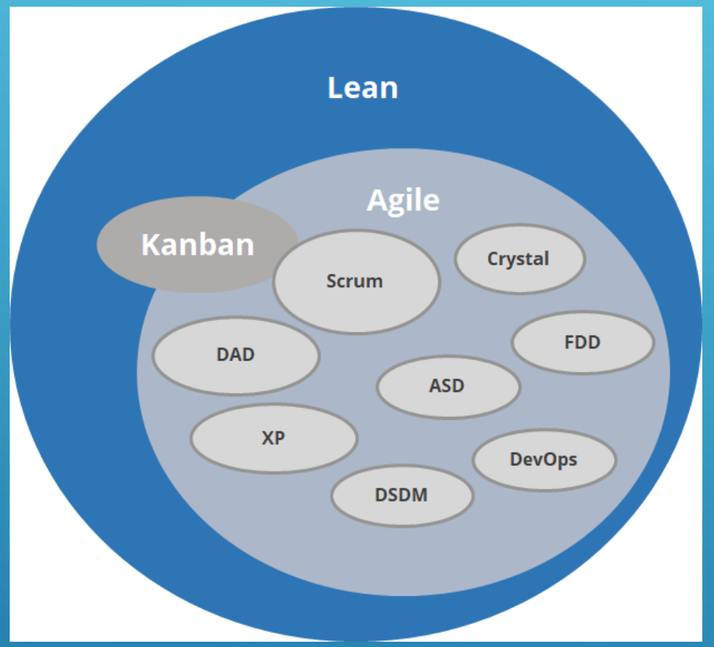
Flow-based approach

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.

Agile Frameworks

AGILE FRAMEWORKS

There is a family of frameworks that are generally recognized as being agile. However, some are only applicable to IT situations.



Tip

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

AGILE FRAMEWORKS

Term	Brief Description
Adaptive Software Development (ASD)	(IT only) Iterative development process
Crystal	(IT only) Iterative development method
Disciplined Agile Delivery (DAD)	(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

Table 2.1 The most well-known agile methods and approaches

AGILE FRAMEWORKS

Term	Brief Description
Dynamic Systems Development Method (DSDM) or 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
Feature-driven development (FDD)	(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
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)

Table 2.1 The most well-known agile methods and approaches

AGILE FRAMEWORKS

Term	Brief Description
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
Scaled Agile Framework (SAFe)	(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
eXtreme Programming (XP)	(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

Table 2.1 The most well-known agile methods and approaches

AGILE BEHAVIORS, CONCEPTS, AND TECHNIQUES

There are a variety of behaviors, concepts, and techniques that are seen as being part of the agile way of working.

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

Table 2.2 Typical agile behaviors, concepts and techniques

AGILE BEHAVIORS, CONCEPTS, AND TECHNIQUES

PRINCE2 and PRINCE2 Agile do not favor one agile approach over another. This is sometimes referred to as being *agile agnostic*.

PRINCE2

PRINCE2 Agile

They can engage with agile in all of its forms to provide a holistic project management approach that can be tailored to suit a wide variety of conditions and working environments.

Kanban and the Kanban Method

KANBAN

Kanban systems are visual management systems that limit the number of work items in circulation. This creates a *pull system*.

Kanban boards have become commonplace when working in an agile way.

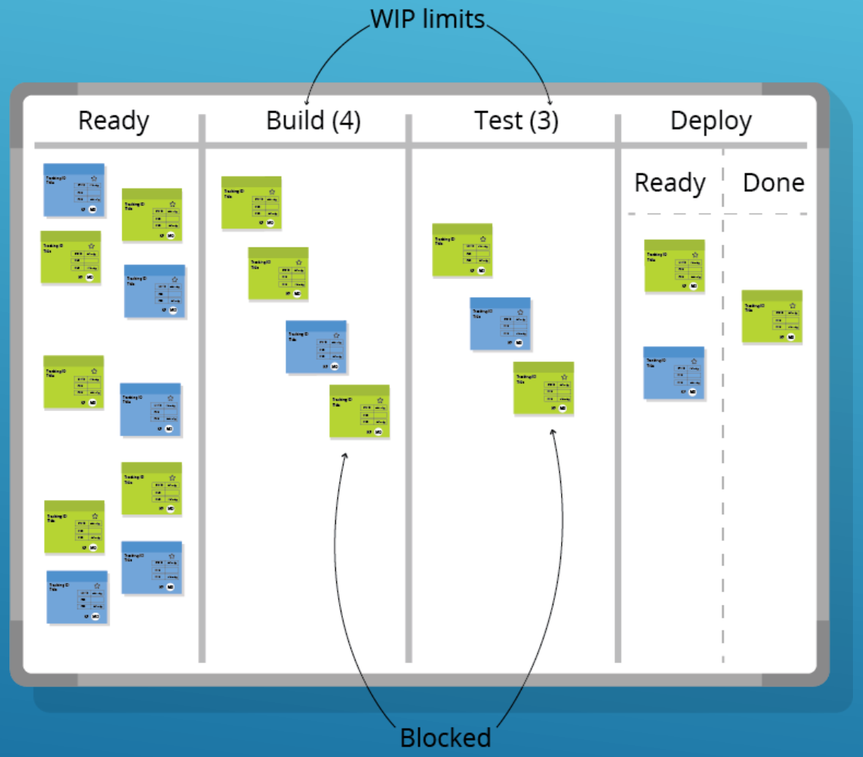


Figure 20.2 An example of how a Kanban board might look

KANBAN METHOD: TOYOTA

In the late 1940s, Taiichi Ohno employed a system of signal cards to deliver the just-in-time element of the Toyota production system.

At Toyota, Kanban is applied to improve flow in the short term and create long-lasting and ongoing change to the underlying processes of the organization.

KANBAN METHOD: APPLICABILITY

The first foundational principle of Kanban method is *start with what you do now.*

Kanban should be seen as a way to increase agility through:

- Improved day-to-day decision-making
- The deferral of commitment
- Reduced lead times
- Increased opportunity for feedback

Work in Progress (WIP)

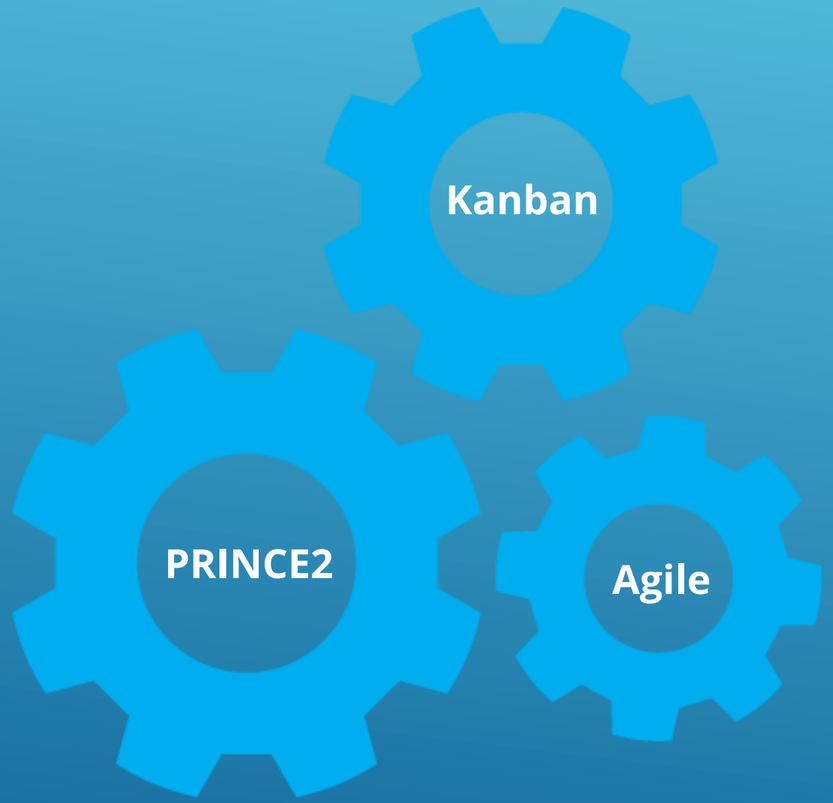
It is the work that has been started but not yet delivered from the system or timebox. It can also indicate the status for incidents, problems, changes, etc.

KANBAN METHOD: APPLICABILITY

Kanban becomes applicable when a reasonably repeatable workflow is established.

In a PRINCE2 context using agile, this is likely to be found after the project initiation document (PID) has been approved and there are discrete work items that can be pulled into a Kanban system.

KANBAN METHOD: APPLICABILITY



- When using any Kanban concept with PRINCE2 and agile, remember at all times that this is in a project context.
- Many Kanban examples relate to solving wider organizational problems or typical BAU scenarios.
- Within a PRINCE2 context, many of the Kanban concepts help to create a more agile environment for PRINCE2.

THE BASICS OF KANBAN

- Visualize
- Limit WIP
- Manage the flow
- Make policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

THE BASICS OF KANBAN

- **Visualize**
- Limit WIP
- Manage the flow
- Make policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

By making work visible, teams can easily see how the work is progressing, what has been done, what is pending, and what the problems are.

A Kanban ticket can exist in many forms.

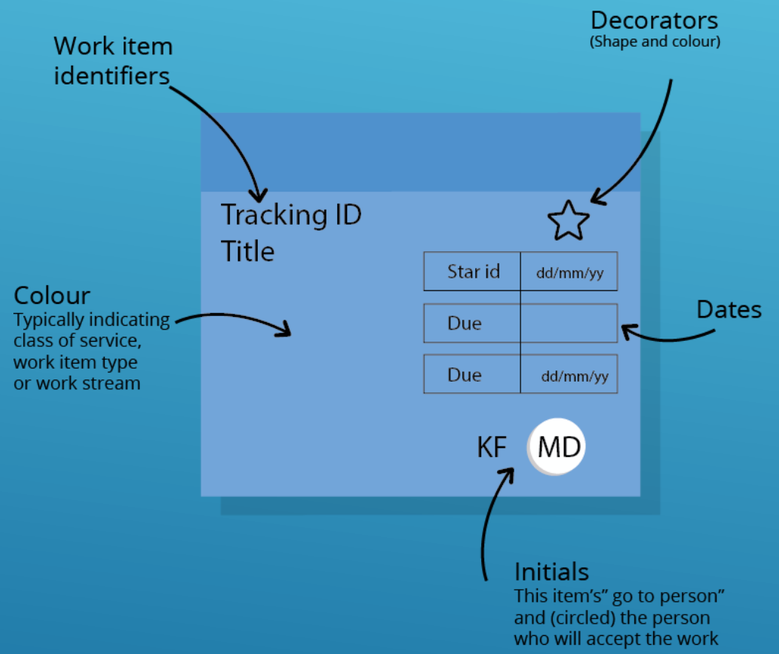


Figure 20.3 An example of how a Kanban ticket might look

THE BASICS OF KANBAN

- **Visualize**
- Limit WIP
- Manage the flow
- Make policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

- The information is recorded on cards or sticky notes that are updated throughout the day.
- *Swim lanes* can be added to identify similar types of work or *classes of service*.

Class of Service

It is a broadly defined category for different types of work. The classes influence selection decisions because different classes of service are typically associated with qualitatively different risk profiles, especially with regard to schedule risk and the cost of delay. Four generic classes of service that are widely recognized are standard, fixed date, expedite, and intangible.

THE BASICS OF KANBAN

- Visualize
- **Limit WIP**
- Manage the flow
- Make policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

It appears counterintuitive to many who think that it may slow the work.

Therefore, understand the reasoning behind limiting WIP as it triggers many events and solves several problems using two analogies:

Reducing the pressure

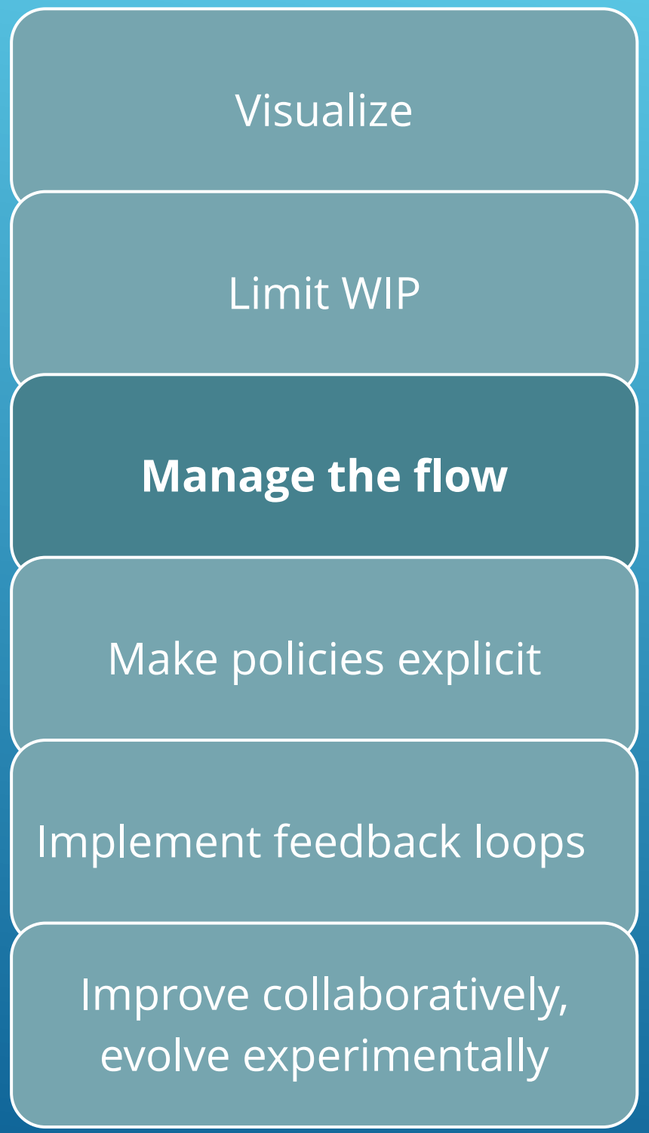
Reducing task-switching

THE BASICS OF KANBAN

- Visualize
- **Limit WIP**
- Manage the flow
- Make policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

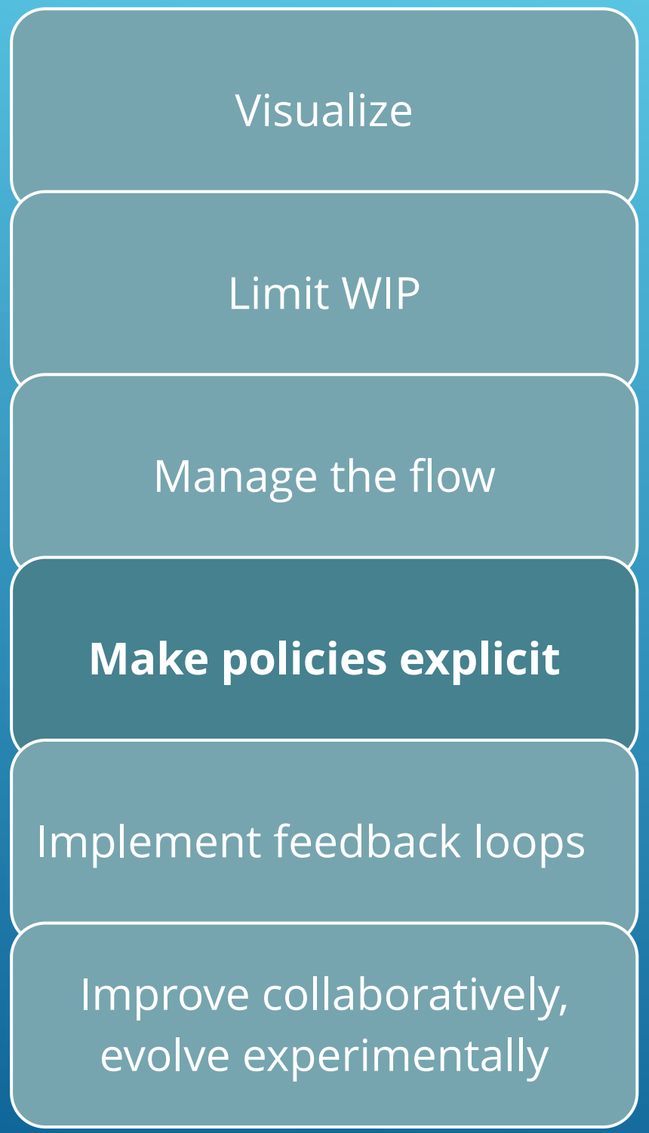
- It denotes the maximum number of sticky notes or cards that are allowed to be present in that column at any one time.
- It underpins the *pull* system which characterizes the way Kanban avoids scheduling work at specific times referred to as a *push system*.
- It reduces the impact of task-switching and multitasking.
- It produces the visual signals that indicate that work can safely be pulled into a place that has the capacity to deal with it effectively.

THE BASICS OF KANBAN



- A Kanban system aims to achieve the highest level of performance from the existing way of working to deliver something of value as quickly as possible.
- Kanban highlights problems that the team needs to solve.
- The Kanban board visualizes the work moving through the system.
- The board acts like a dashboard which enables the team to see blockers and areas where the flow is not smooth.

THE BASICS OF KANBAN



- The teams need clearly defined boundaries to operate though there is empowerment, self-organization, and trust.
- The team needs to clearly define how it works and make the policies transparent.
- The policies should evolve and be built collaboratively over time to create a set of guidelines.

THE BASICS OF KANBAN

Visualize

Limit WIP

Manage the flow

Make policies explicit

Implement feedback loops

Improve collaboratively, evolve experimentally

- Consumers judge the value delivered by any process.
- Assessing the feedback quantitatively is advantageous as it directly affects the subsequent deliveries.
- Shortening the feedback loop to ensure that the most valuable work is in the Kanban system is essential to deliver the most value.
- The Kanban method includes four types of review to gather feedback:
  - The stand-up meeting
  - The service delivery review
  - The operations review
  - The risk review

THE BASICS OF KANBAN

- Visualize
- Limit WIP
- Manage the flow
- Make policies explicit
- Implement feedback loops
- **Improve collaboratively, evolve experimentally**

- The Kanban method embraces the idea that improvement is a collaborative exercise.
- The transparency and the ease at which the Kanban system can be modified creates the natural conditions for collaborative improvement to occur.
- The team can form hypotheses of what may be holding the system back and then agree to changes that can be tested in a safe-to-fail manner.

Safe-to-fail

A safe-to-fail experiment is designed to have only limited impact on the system or the plan in the event of failure.

Scrumban

SCRUMBAN

Scrum and Kanban are two of the most popular agile approaches. They are similar as they focus strongly on process improvement, transparency, and empiricism.

Scrum

- Has specific roles
- Work is timeboxed
- Work is related to a specific product
- Cannot be applied to Kanban environment

Kanban

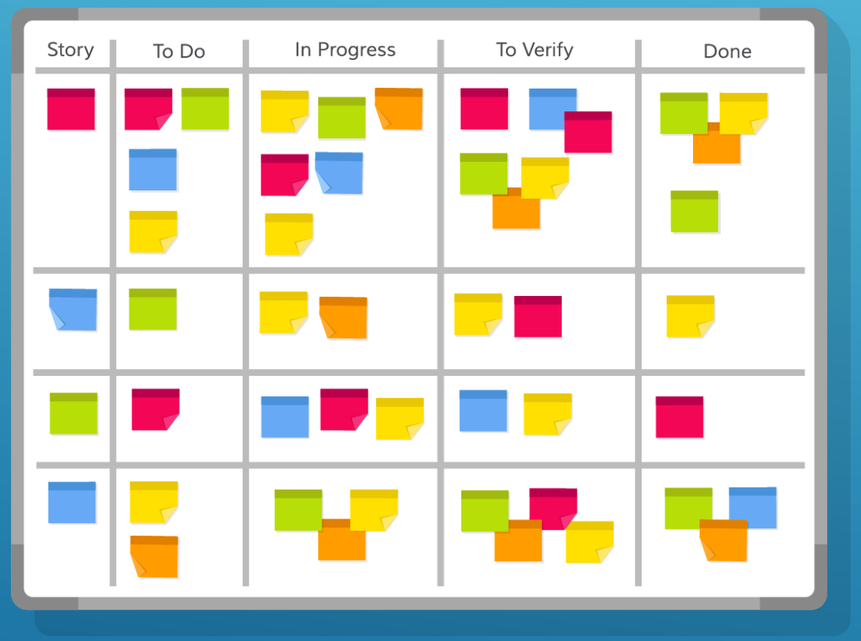
- Has no defined roles
- Work is pulled to create a flow
- Work may relate to anything
- Can be applied to Scrum environment

SCRUMBAN

- Scrumban is the application of Kanban, where the underlying process is based on Scrum.
- This may involve the use of Kanban systems to manage the work of the sprint.
- It is more powerful to apply Kanban to a broader workflow that starts upstream of the build process and ends with customer delivery or even post-deployment validation.

WORK ITEM SIZE AND SIMILARITY

Kanban systems are able to deal with multiple types of work or classes of service.



- The flow will be more predictable if work items are within the same order of magnitude in size, complexity, or risk.
- It is achieved by policies on work item size and adjusted where necessary for risk.
- Teams learn to recognize their disproportionate risk and develop the skills to identify and deliver value in smaller work items.

EXPERIMENTS

When a team looks to improve how the system works to deliver more value to the customer, it should do it in a controlled and objective way.

The Kanban method recommends using the *scientific method* to achieve this.

Scientific method is a technique for improving understanding and knowledge by going through a process of several steps such as:

- Ask a question
- Carry out research
- Create a hypothesis
- Carry out experiments
- Analyze the results
- Draw a conclusion

CUMULATIVE FLOW DIAGRAM (CFD)

CFD is a common technique used in Kanban to track work items, which shows the amount of work in each column on a daily basis.

Column counts

Day	Ready	Build	Test	Ready to deploy	Deployed
8	2	4	4	1	0
9	4	5	3	0	3
10	3	3	5	1	3
11	2	3	4	3	3
12	4	3	4	0	8
13	3	2	5	1	8
14	7	2	4	3	8
15	7	2	4	0	13
16	7	2	5	0	13
17	5	3	5	0	14
18	5	3	4	0	15
19	4	2	5	0	16
20	3	2	3	0	19
21	3	1	1	0	22

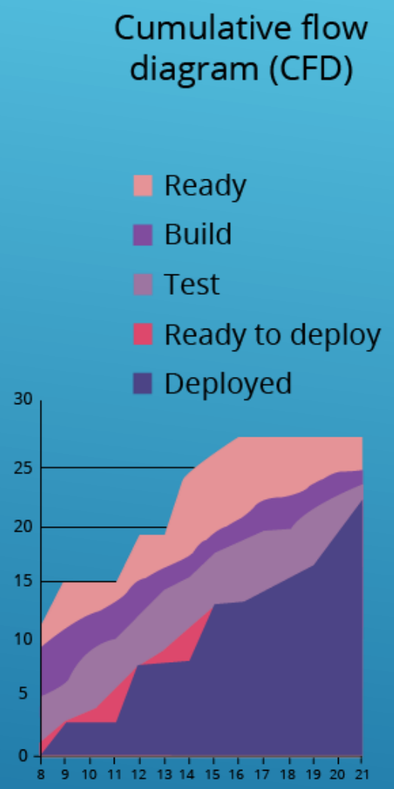


Figure 20.4 A cumulative flow diagram

CUMULATIVE FLOW DIAGRAM (CFD)

WIP is the vertical difference between the line showing work that is ready and the line showing what has been deployed, whereas lead time is the difference horizontally between the two.

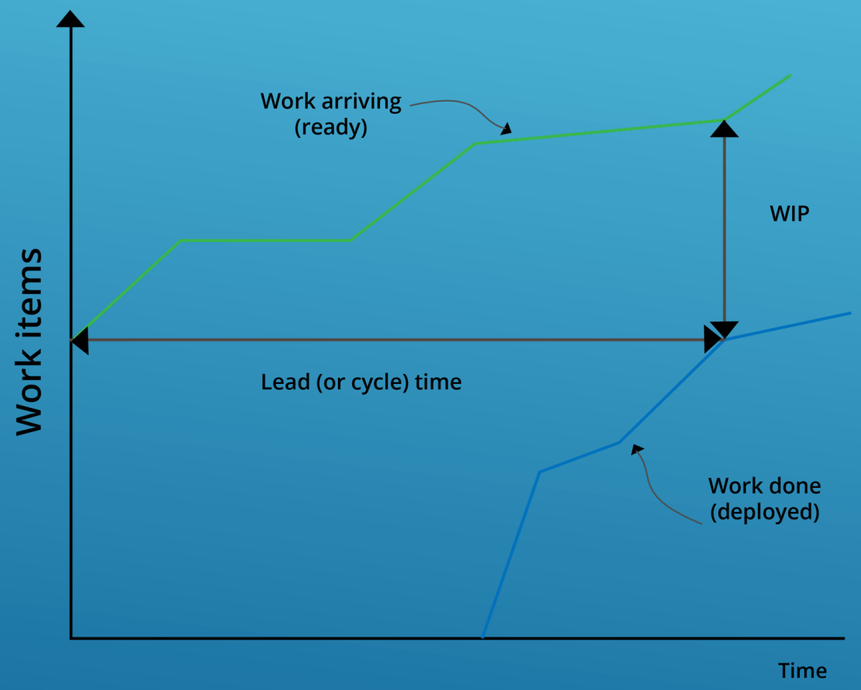


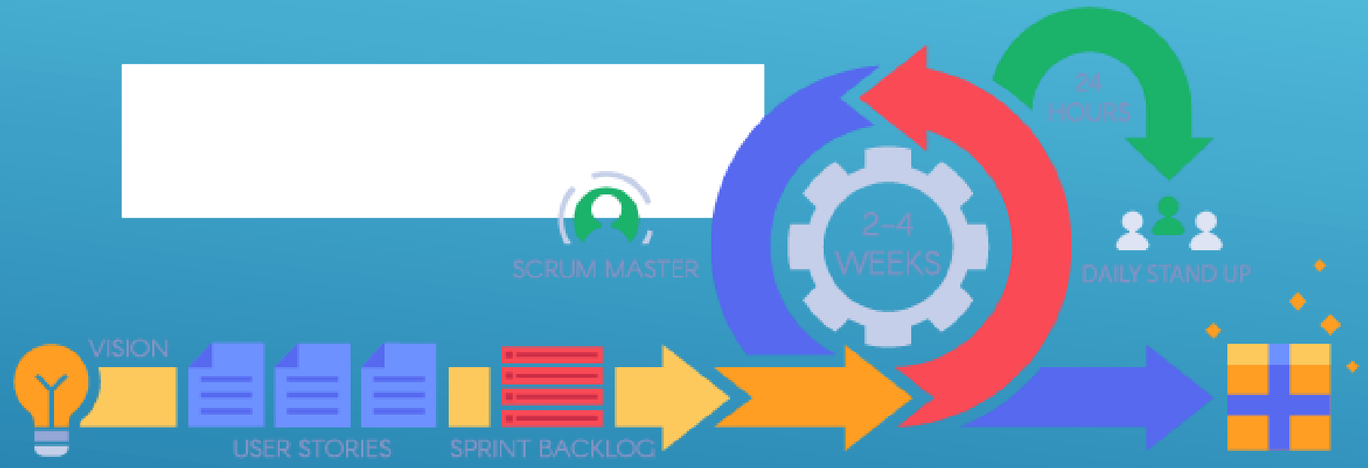
Figure 20.5 A cumulative flow diagram

Lead time or cycle time

These two terms are interpreted differently by many in the Kanban community. They refer to how long a work item takes to go through the system or timebox. So although they are often interpreted differently, they are, in effect, the same thing.

HINTS THAT MAY PROVE USEFUL

See what Kanban is and what it offers and then apply it to a project context in the most appropriate way



- A two-week sprint could be planned in advance with a finite amount of work as per Scrum
- Alternatively, the sprint could be unplanned and work could be pulled from a list when necessary

HINTS THAT MAY PROVE USEFUL

Improve flow and deliver value as early as possible to keep with the thinking behind flexing what is being delivered. Kanban aims for timeliness and reducing the impact of *cost of delay.*

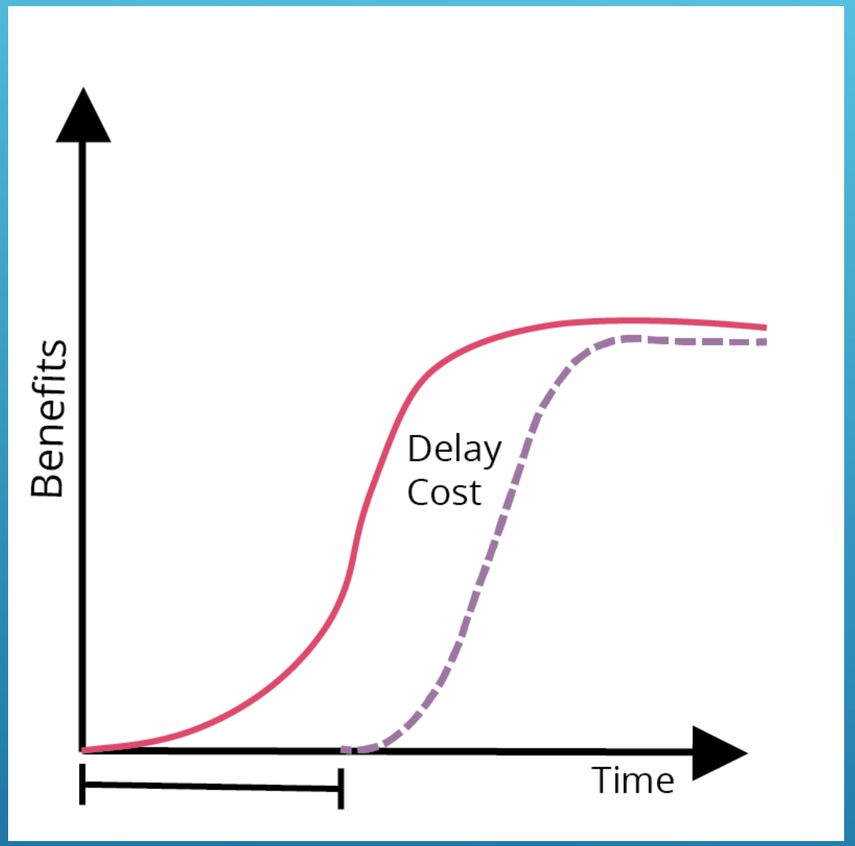


Figure 20.6 The effect of delaying the delivery of a product

HINTS THAT MAY PROVE USEFUL

Little’s law is part of the queueing theory body of knowledge. An adjusted version of it is used to understand the flow of work through a Kanban system.

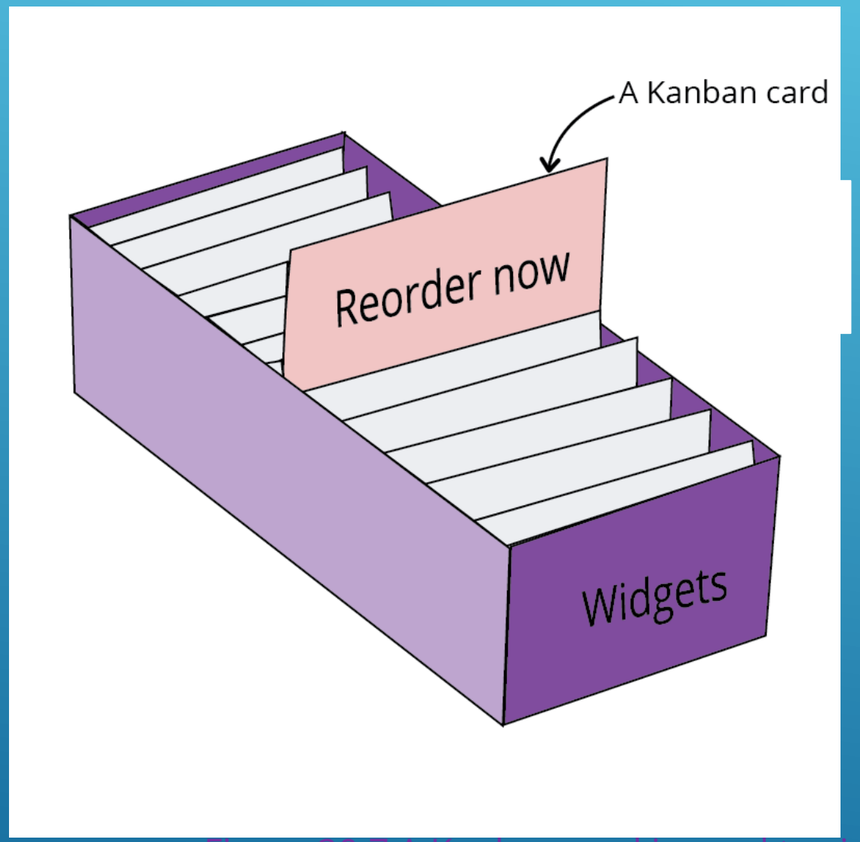
Little’s Law

$L = \lambda W$

In simple terms, it is the average number of items in a system. L is equal to the average arrival rate, λ, multiplied by the average time an item spends in the system, W (assuming that this is over a long enough period of time and the system is stable).

KANBAN: ORIGIN

The word Kanban is used in both Japanese and Chinese, though with different meanings.



- **Japanese**: Signal card or sign or visual board
- **Chinese:** Looking at the board

Figure 20.7 A Kanban card is used to signal that stock needs to be replenished

The Lean Startup Method

LEAN STARTUP

It is a method to grow new businesses, and develop existing ones, through product innovation in uncertain markets.

The core concepts of Lean Startup that apply to PRINCE2 are:

- Build, measure, and learn
- Create a minimum viable product (MVP)
- Fail fast
- Validated learning

LEAN STARTUP

Lean Startup works to create a simple approach that can be applied to any situation where uncertainty exists, such as a project.

It can be used partially or wholly as a source of techniques as it is like PRINCE2 and agile in that it is product-focused and responsive to change.

LEAN STARTUP

For the customers, time is important and they want as much as they can get in as short a time as possible.

Startup companies using new technologies have to use a different management approach.
The approach needs to be agile, based on the early delivery of value.

MANAGE UNCERTAINTY

There is a need for a managed process though Lean Startup is geared to handle uncertainty or innovate.

Overplanning and forecasting too far ahead wastes effort, so does a *just do it* approach.

MANAGE UNCERTAINTY

Some of the core concepts and techniques in Lean Startup are fundamental to the most effective way to deliver a product at the end of a timebox or a project.

Lean Startup focuses on uncertainty, learning, and handling change.

APPLY LEAN STARTUP TO PRINCE2

Applying Lean Startup to PRINCE2 should be seen in the context of a timebox.
This timebox could relate to the whole project or just a two-week sprint.

Lean Startup aims at a group of people and creates a product where there's uncertainty.
Therefore, Lean Startup is useful to PRINCE2 as a technique in this context.

APPLY LEAN STARTUP TO PRINCE2

The core idea of Lean Startup is that there is a need to *focus on learning* to be successful.
It is vital to understand the customer's needs quickly.

Lean Startup refers to this as shortening or accelerating the feedback loop.
This is in accordance with the PRINCE2 Agile behavior of exploration.

MEASURES AND VALIDATED LEARNINGS

Measurable feedback is essential to learning. Even if the feedback is subjective, it has to be measurable so that it can be quantified.

- Lean Startup refers to *vanity metrics* and *actionable metrics*.
- You need to capture the metrics that are directly related to the business case or a timebox objective.
- These are actionable metrics and not vanity metrics.
- The vanity metrics do not relate directly to the business case.

MEASURES AND VALIDATED LEARNINGS

If a tourist attraction is looking to increase revenue, then:

- The revenue received during a day is an actionable metric
- The number of daily visitors is a vanity metric

MEASURES AND VALIDATED LEARNINGS

- How a project is planned has a direct effect on how feedback is received.
- An early release of a part of the product into operational use will provide feedback.
- This feedback could be negative and result in the project being canceled.
- In Lean Startup, if you are going to fail you need to fail as fast as possible.

MEASURES AND VALIDATED LEARNINGS

- A company took 6 months to build a product and when they launched it the product failed.
- If they had released a reduced version of the product after one month they would have failed 5 months earlier and saved a lot of money.
- The key point here is that the loss of five months' money is not as important as the loss of five months of learning.

The same applies to a two-week sprint. It may be early in a sprint that a prototype is made, shown to a customer, and rejected immediately. The learning has already started.

BUILD, MEASURE, AND LEARN

The three steps of build, measure, and learn apply both to releases and interim products. The most important of the three is the final step to do with learning.

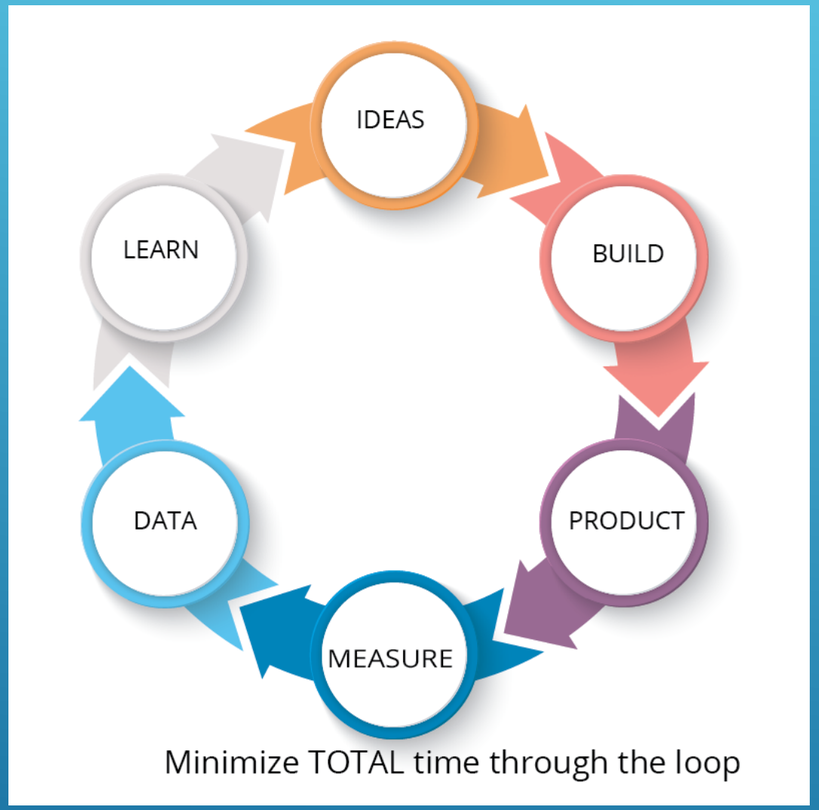


Figure 20.8 The build–measure–learn feedback loop from Lean Startup

MINIMUM VIABLE PRODUCT (MVP)

- MVP is the simplest form of the product to get feedback.
- It involves a limited set of features that can be enhanced throughout the project.
- What constitutes an MVP for a project depends on the levels of uncertainty involved.
- In very innovative situations, it can involve educated guesswork or instincts.
- The Lean Startup method forces target measures to be created and validated as soon as possible by the results.

MINIMUM VIABLE PRODUCT (MVP)

- Lean Startup assesses *minimum viability* based upon *what is the least that can be done to learn*.
- The team needs to learn as fast as possible or *learn the most with the least effort.*
- A common agile view of MVP is about the commercial viability of the product in terms of whether or not it will sell.
- PRINCE2 Agile does not share this view and defines the MVP based on the Lean Startup definition.

MINIMUM VIABLE PRODUCT (MVP)

Minimum Viable Product

In a PRINCE2 Agile context, the term MVP broadly aligns with the Lean Startup view that it is a *version of the final product which allows the maximum amount of validated learning with the least effort*.

This should not be confused with the viability of the project as a whole. Typically, an MVP would be delivered as early as possible during the project.

It is important to note that an MVP is about learning; it may not go into operational use and may be in the form of a simple experiment or prototype.

KAIZEN AND CONTINUAL IMPROVEMENT

Lean Startup sees the need:

- For process when trying to be agile
- To continually improve that process
- For hard data to scientifically evaluate feedback and learnings

Being flexible and dynamic needs control. *Adhocracy* will rarely work even in the volatile start-up arena.

Therefore, the concepts of Lean Startup can be used to complement PRINCE2 as it believes that to be responsive you need control.

HINTS THAT MAY PROVE USEFUL

Where there is extreme uncertainty, Lean Startup is happy for the MVP to have less than the ideal level of quality.

In practice, it is said “Let’s not guess the quality level, let’s find out from the customer’s feedback.”

This level of quality will need to be defined from the start as it does not represent the quality level being compromised.

HINTS THAT MAY PROVE USEFUL

Lean Startup prefers to segment the feedback it receives by groups of users or *cohorts*.

This illustrates why it is important to engage with a representative view of the stakeholders from the customer side.

A single product owner may be a disadvantage if you are using segmentation with cohorts approach.

HINTS THAT MAY PROVE USEFUL

Lean Startup refers to *funnel metrics.*

These represent the data that can be learned to an extent, but the key metrics need to tie back to the business case to validate the original rationale.

On a PRINCE2 project using agile, these funnel metrics may start arriving during the project and may affect how future work is planned and organized.

REDUCE UNCERTAINTY

Lean Startup is at its best when faced with extreme uncertainty and, in a project context, this level of uncertainty may only apply to a minority of situations.

However, uncertainty will always exist but the goal remains the same in all situations.

Workshops

WORKSHOPS

A workshop is an activity where several people come together to achieve an objective by harnessing the interactions and creativity of the participants.

- A workshop would last from two or three hours to a whole day.
- A neutral facilitator, who has no stake in the outcome, runs the workshop.

In the workshop technique, it is better to consolidate the understanding of many people by listening to them at the same time rather than listening to them separately.

WORKSHOPS: THE BASICS

In a workshop, different viewpoints may occur when everyone is involved. However, the interaction needs to be managed to ensure that everyone can contribute fairly.

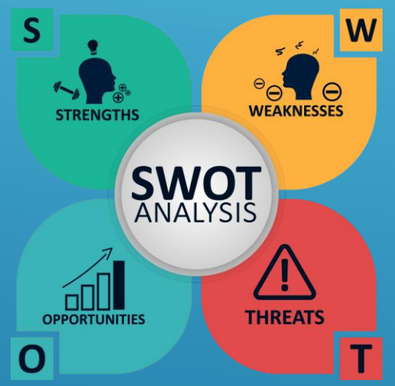
Steps to prepare for a workshop:

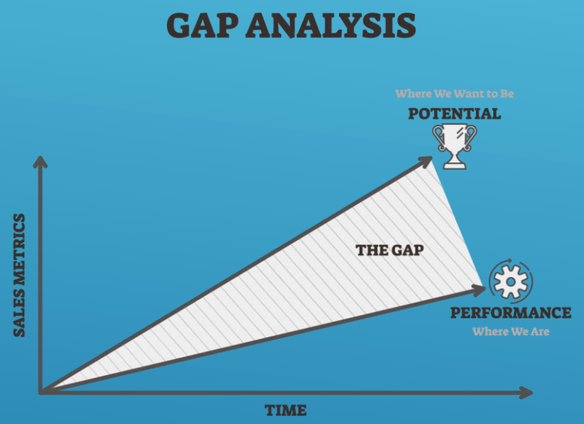
- Workshop objective
- Attendees
- Agenda
- Logistics
- Prereading

An experienced facilitator, familiar with these steps, is preferred to work with the person authorizing the workshop to structure the workshop in the most appropriate way.

WORKSHOPS: THE BASICS

There are a variety of tools and techniques available to address certain problems and situations.





An experienced facilitator would be conversant with many of these tools and techniques.

WORKSHOPS: POSSIBLE TECHNIQUES

Techniques	Description
SWOT analysis	Focuses on the four areas of strengths, weaknesses, opportunities, and threats for a given situation
Impact or effort grids	A two-by-two (four box) grid that allows items to be positioned against two criteria on the x and y axis
Rich pictures	Using visualization to convey messages (often feelings) in a form that can use metaphors and humor
Prioritization with dots	The use of sticky dots or marker pen dots to quickly vote on a set of options
Gap analysis	A three-step technique used to describe how something can get from one state or situation to another

Table 26.1 Possible workshop techniques

WORKSHOPS: POSSIBLE TECHNIQUES

Techniques	Description
Brainstorming	A way of generating ideas, which normally involves sticky notes so that all ideas are initially produced without being affected by other people
Visioning	Creating shared goals or objectives, often using visualization. Defining the overall *why*
The five whys (repeatedly asking *why)*	A questioning technique to get to the root of a problem or request
Dr Edward de Bono's Six Thinking Hats	A technique to help people think in six different ways: • The White Hat calls for information known or needed • The Yellow Hat symbolizes brightness and optimism • The Black Hat is judgment • The Red Hat signifies feelings, hunches, and intuition • The Green Hat focuses on creativity • The Blue Hat is used to manage the thinking process

Table 26.1 Possible workshop techniques

WORKSHOP TECHNIQUE: EXAMPLES

Group work

- It may be a good idea to break the whole group into smaller sub-groups.
- This can enable more areas to be covered.
- This allows for quieter members of the group to contribute more freely.

Sticky notes

- It provides many advantages such as:
  - Make people concise
  - Is somewhat anonymous
  - Help the group create output quickly
  - Is movable and visual

USING WORKSHOPS

Workshops can be used whenever needed and at any point in a project.

The workshop technique can be useful in situations such as:

- Planning and estimating
- Reviewing
- Problem solving
- Requirements gathering
- Project kick-offs
- Carrying out a stakeholder analysis
- Identifying and analyzing risk

HINTS THAT MAY PROVE USEFUL

- Groups using workshops without a facilitator must have established and agreed on its own working norms.
- A workshop event takes a lot of time and resources to set up and run.
  - It is advisable to question whether a workshop is really necessary or is there another way of achieving the objective.
- With some workshops, it is a good idea to create something collectively as a group.
- At other times, it is better for specific individuals to create something and then have it reviewed as a group.

RUN WORKSHOPS WELL

Workshops, when run correctly, can create high-quality outputs in short spaces of time through motivated individuals collaborating and communicating effectively.

This in turn creates clarity, consensus, and ownership.

Transitioning to Agile

TRANSITIONING TO AGILE

Most organizations make a mistake by seeing working in an agile way as a goal as opposed to seeing it as an enabler to help the organization achieve its goals.

- Organizations should understand the problems before commencing on any transition to a new or different way of working.
- Organizations can encounter many common problems in *traditional* projects.

Therefore, when using PRINCE2 and agile together to solve these problems, create a baseline of where the organization currently is before starting on a journey of transition.

FORMS OF SUCCESS

When using PRINCE2 and agile on a project, distinguish between the following three different types of success:

1. The success of the business case in terms of the benefits delivered
2. The success of the project and the project management involved
3. The success of the agile way of working

Benefits are always important. Different stakeholders may have their own measures of success depending on what benefits they stand to gain from the project.

FORMS OF SUCCESS

- It is important to understand the relative merits of these three areas.
- It is important that all of these three areas are measured.
- It is important that there are no overlaps or blurring of boundaries between the areas

FORMS OF SUCCESS

Although there are problems using PRINCE2 or agile with the project, the customer may still be happy if the product produced sells well and achieves the benefits that were forecast.

Conversely, even if PRINCE2 and agile enable the project to come ahead of time and under budget, the customer is unlikely to be happy if the final product does not sell as well as expected.

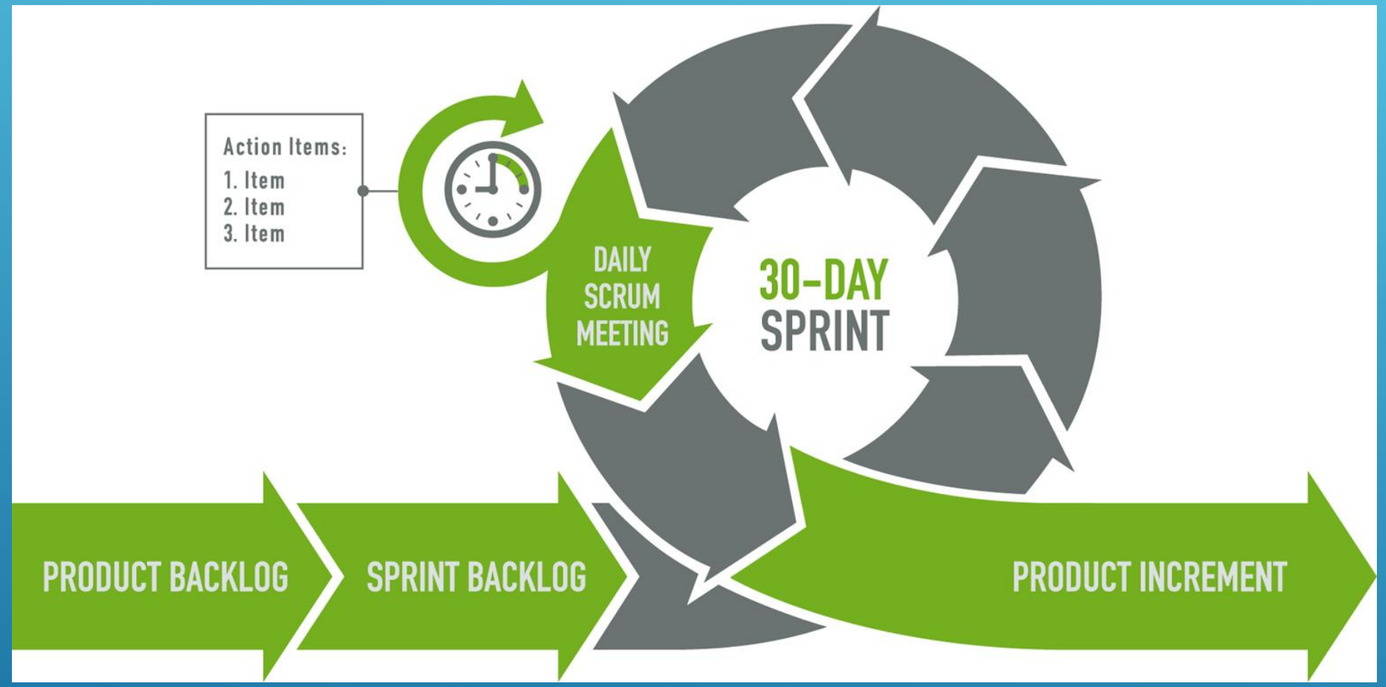
FORMS OF SUCCESS

- Assess the merits of that investment when an organization invests in improving how projects are delivered
- Distinguish between the benefits delivered by the appropriate use of PRINCE2 and the benefits delivered by agile
- Measure the contribution of PRINCE2, agile, and PRINCE2 with agile as they are all enablers

The Scrum Guide

SCRUM

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.



It is lightweight, simple to understand, and difficult to master.

SCRUM

- It is a process framework that has been used to manage work on complex products since the early 1990s.
- It is a framework within which one can employ various processes and techniques.
- It makes clear the relative efficacy of product management and work techniques.
- It proved effective in iterative and incremental knowledge transfer.
- The essence of Scrum is a small team of people. The individual team is highly flexible and adaptive.

SCRUM FRAMEWORK

- It consists of scrum teams and their associated roles, events, artifacts, and rules.
- Each component within the framework serves a specific purpose and is essential to Scrum's success and usage.
- The rules of Scrum bind together the events, roles, and artifacts, governing the relationships and interaction between them.

USES OF SCRUM

Scrum was initially developed for managing and developing products. Starting in the early 1990s, Scrum has been used extensively, worldwide, to:

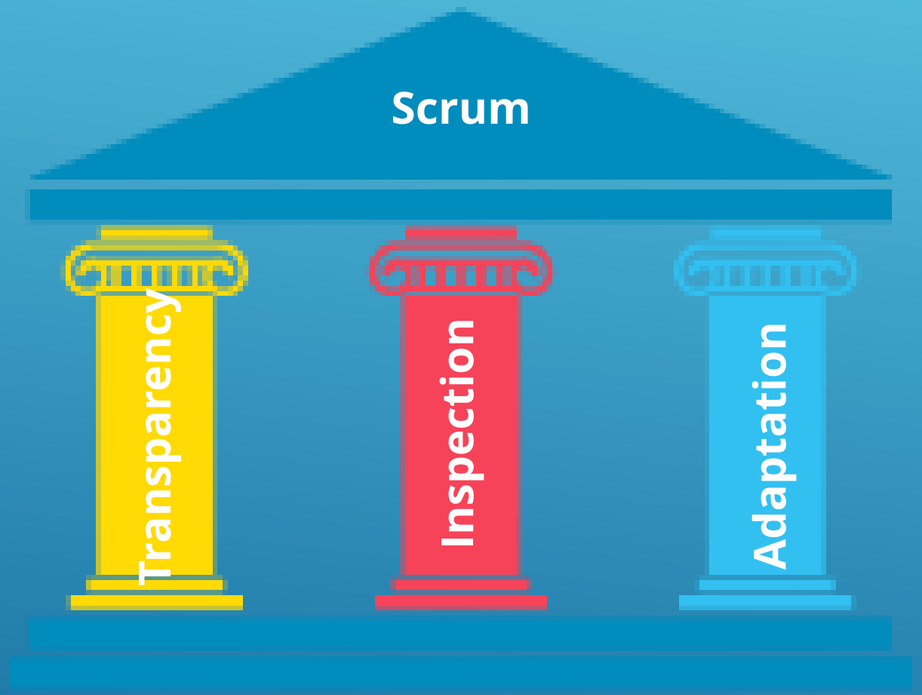
Scrum

Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

1. Research and identify viable markets, technologies, and product capabilities
2. Develop products and enhancements
3. Release products and enhancements, as frequently as many times per day
4. Develop and sustain cloud and other operational environments for product use
5. Sustain and renew products

SCRUM THEORY

Scrum is founded on empirical process control theory or empiricism.
Empiricism asserts that knowledge comes from experience and making decisions on what is known.

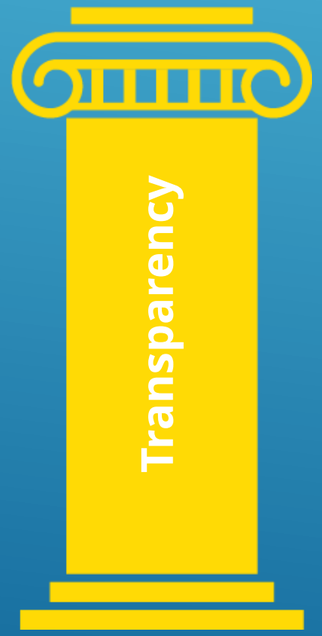


The three pillars that uphold every implementation of empirical process control.

TRANSPARENCY

Significant aspects of the process must be visible to those responsible for the outcome.

Transparency requires those aspects to be defined by a common standard so observers share a common understanding of what is being seen.



Example:

- A common language referring to the process must be shared by all participants.
- Those performing the work and those inspecting the resulting increment must share a common definition of *Done*.

INSPECTION

Scrum users must frequently inspect Scrum artifacts and progress toward a sprint goal to detect undesirable variances.

- Inspection should not be so frequent that inspection gets in the way of the work.
- Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

ADAPTATION

If an inspector determines that one or more aspects of a process deviate outside the acceptable limits, adjust the process or the material being processed.

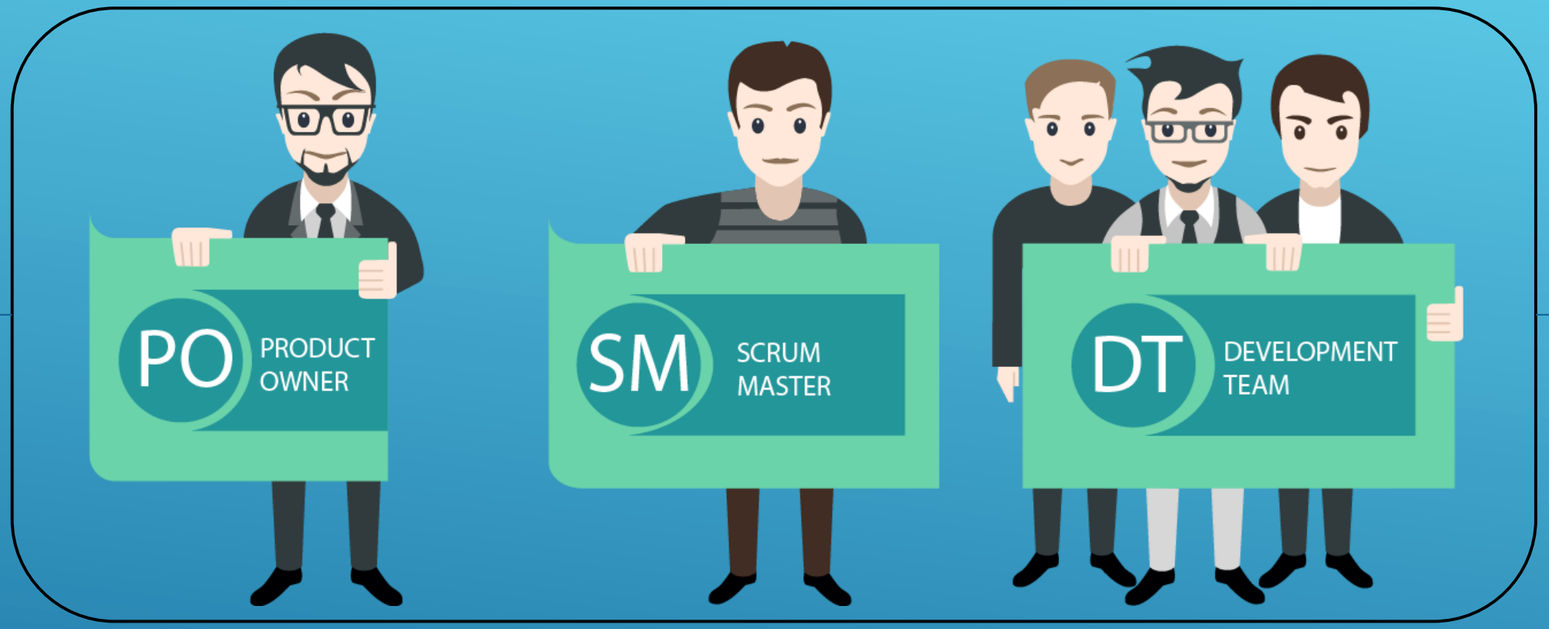
An adjustment must be made as soon as possible to minimize further deviation.

Scrum prescribes four formal events for inspection and adaptation, as described in the Scrum events:

- Sprint planning
- Daily Scrum
- Sprint review
- Sprint retrospective

THE SCRUM TEAM

The scrum team consists of three roles and are self organizing and cross-functional.

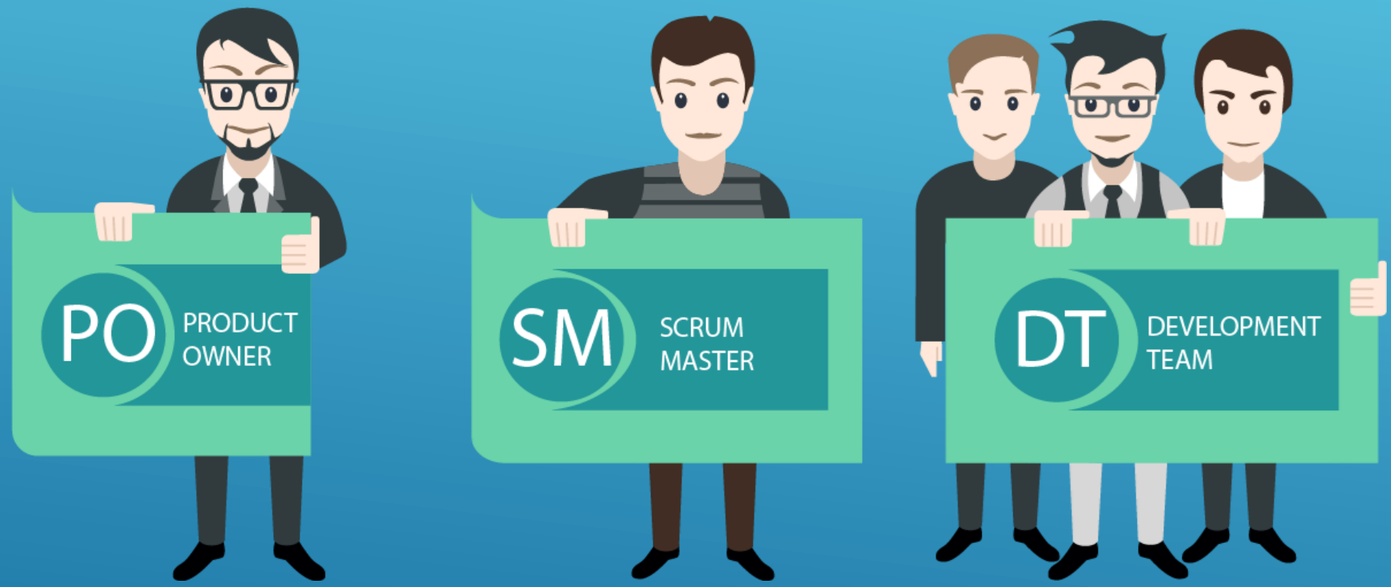


Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team.

Cross-functional teams have all competencies needed to accomplish the work without depending on others.

THE SCRUM TEAM

The team model in Scrum is designed to optimize flexibility, creativity, and productivity.



Scrum teams deliver products iteratively and incrementally, maximizing opportunities for feedback.

THE PRODUCT OWNER

The product owner is responsible for maximizing the value of the product resulting from the work of the development team.

- The product owner manages the product backlog.
- The product owner is one person and not a committee.
- Those wanting to change a product backlog item's priority must address the product owner.
- For the product owner to succeed, the entire organization must respect their decisions.
- The product owner's decisions are visible in the content and ordering of the product backlog.

THE PRODUCT BACKLOG MANAGEMENT

Product backlog management:

- Clearly expresses product backlog items
- Orders the items in the product backlog to best achieve goals and missions
- Optimizes the value of the work the development team performs
- Ensures that the product backlog is visible, transparent, and clear to all
- Shows what the scrum team will work on next
- Ensures the development team understands items in the product backlog to the level needed

THE DEVELOPMENT TEAM

The team consists of professionals who do the work of delivering a potentially releasable increment of *done* product at the end of each sprint.

They are structured and empowered by the organization to organize and manage their own work.

THE DEVELOPMENT TEAM

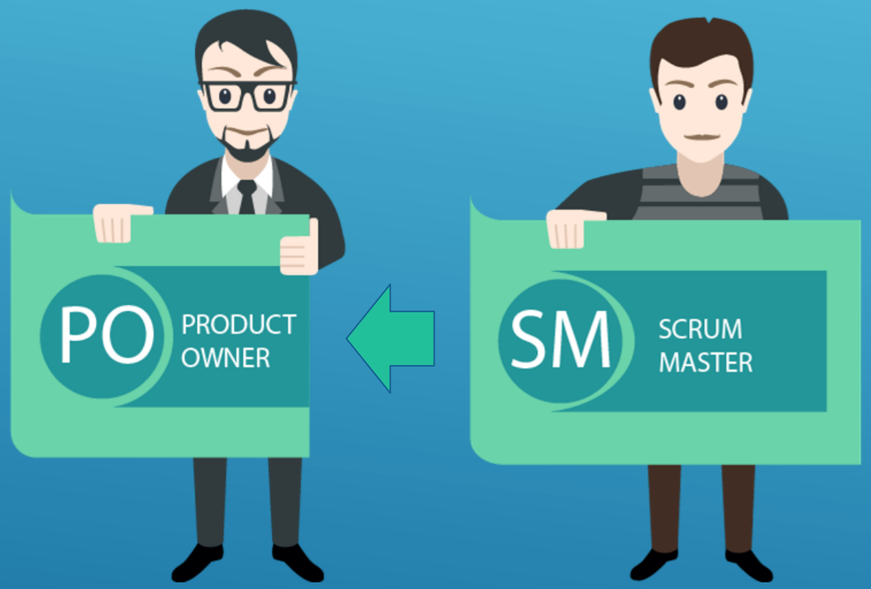
- They are self organized and cross-functional.
- Scrum recognizes no titles for the development team members.
- Scrum recognizes no subteams in the development team.
- Individual development team members may have specialized skills and areas of focus, but accountability belongs to the development team as a whole.
- Optimal development team size is small enough to remain nimble and large enough to complete significant work within a sprint.

THE SCRUM MASTER

The scrum master is responsible for promoting and supporting Scrum. They do this by helping everyone understand Scrum theory, practices, rules, and values.

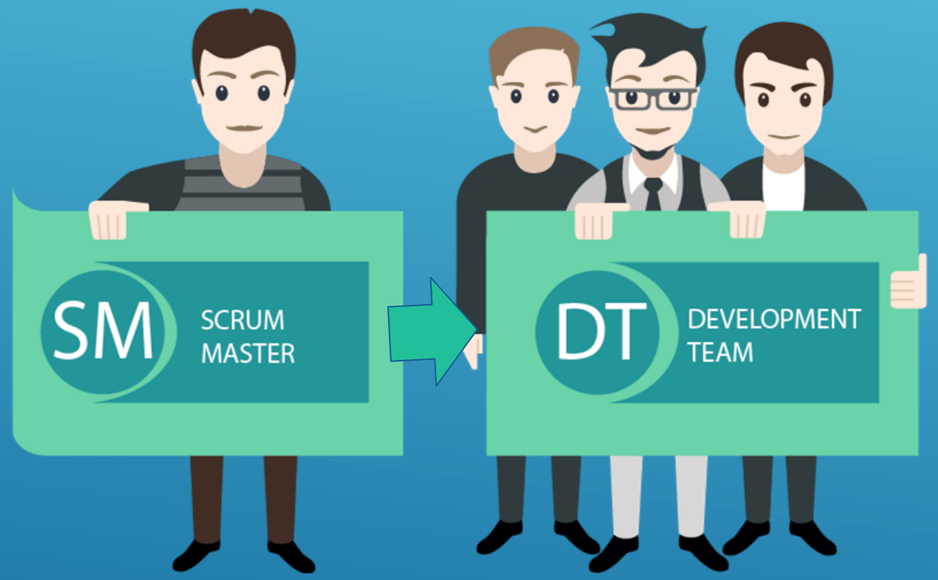
- Is a servant-leader for the scrum team
- Helps those outside the scrum team understand which of their interactions are helpful
- Helps everyone change these interactions to maximize the value created by the scrum team

SCRUM MASTER SERVICE TO THE PRODUCT OWNER



- Ensures that goals, scope, and product domain are understood by everyone on the scrum team
- Finds techniques for effective product backlog management
- Helps the scrum team understand the need for clear and concise product backlog items
- Understands product planning in an empirical environment
- Ensures the product owner knows how to arrange the product backlog to maximize value
- Understands and practices agility
- Facilitates scrum events as requested or needed

SCRUM MASTER SERVICE TO THE DEVELOPMENT TEAM



- Coaches the development team in self organization and cross-functionality
- Helps the development team to create high-value products
- Removes impediments to the development team’s progress
- Facilitates scrum events as requested or needed
- Coaches the development team in organizational environments

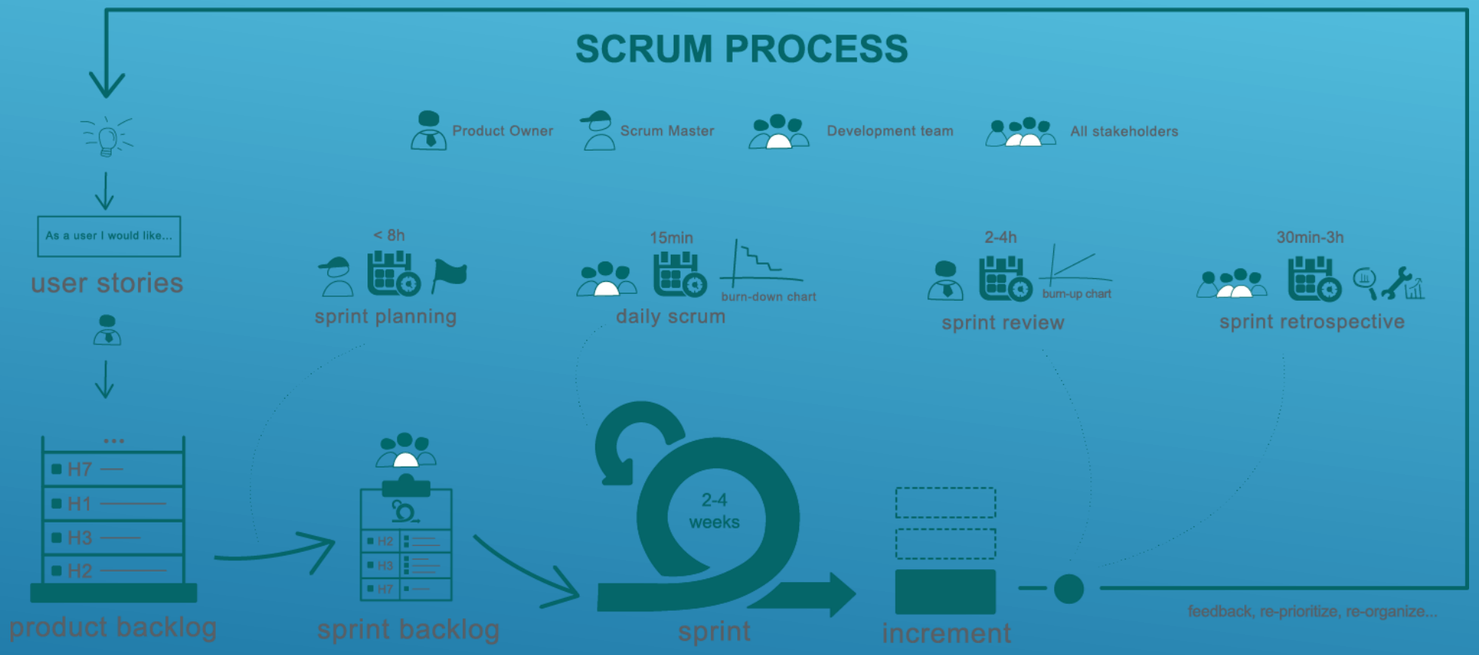
SCRUM MASTER SERVICE TO THE ORGANIZATION

- Leads and coaches the organization in its scrum adoption
- Plans scrum implementations within the organization
- Helps employees and stakeholders understand and enacts scrum and empirical product development
- Causes change that increases the productivity of the scrum team
- Works with other scrum masters to increase the effectiveness of the application of scrum in the organization

Scrum Events and Artifacts

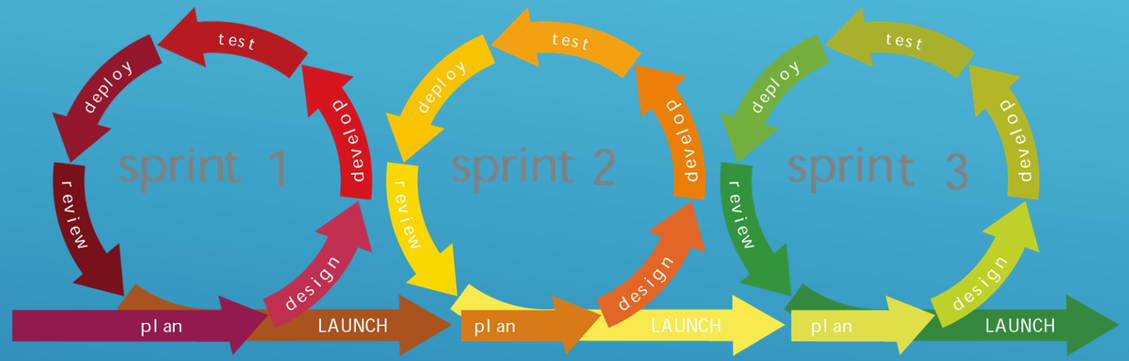
SCRUM EVENTS AND ARTIFACTS

Scrum events are used to create regularity and minimize the need for meetings. These events are timeboxed so that each event has a maximum duration.



Artifacts defined by Scrum represent work or value to provide transparency and opportunities for inspection and adaptation.

SCRUM EVENTS



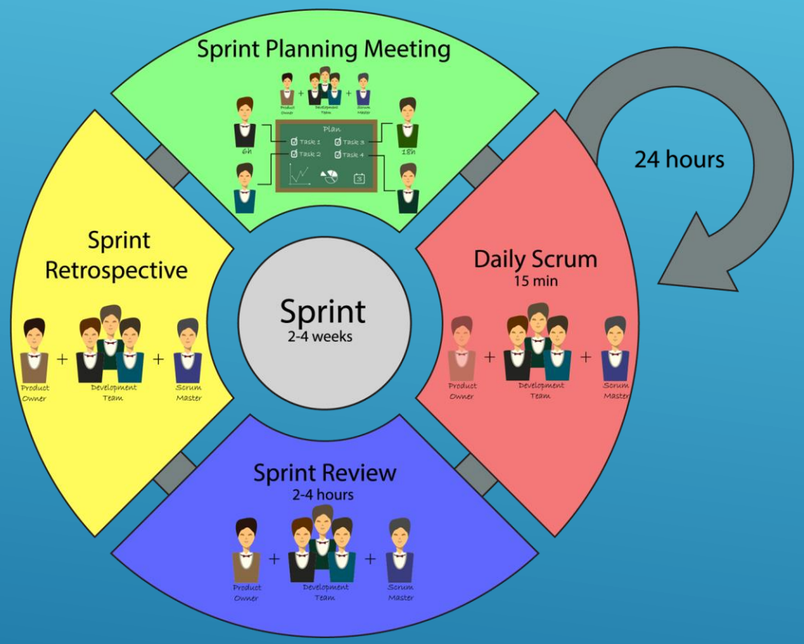
- Once a sprint begins, the duration is fixed and cannot be altered.
- The remaining events may end whenever the purpose of the event is achieved.
- An appropriate amount of time is spent without allowing waste in the process.

Each event in Scrum is a formal opportunity to inspect and adapt something.

Failure to include any of these events results in reduced transparency and is a lost opportunity to inspect and adapt.

THE SPRINT

Sprint is a timebox of one month or less where a *done*, useable, and potentially releasable product increment is created.



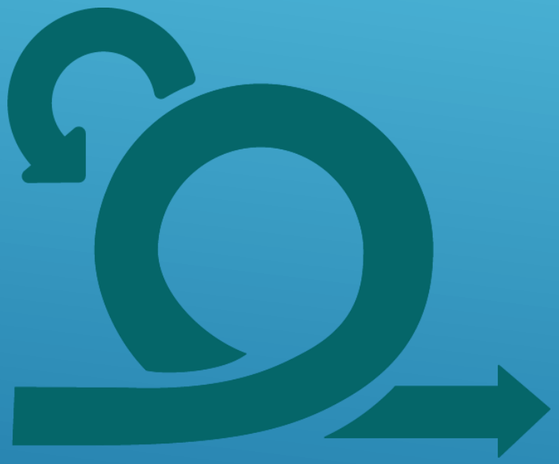
A new sprint starts immediately after the conclusion of the previous sprint.

THE SPRINT

- During the Sprint:
  - No changes are made that would endanger the sprint goal
  - Quality goals do not decrease
  - Scope may be clarified and renegotiated
- Each Sprint may be considered a project with no more than a one-month horizon.
- Each sprint has a goal, a design, a flexible plan, the work, and the resultant product increment.

Sprints enable predictability by ensuring inspection and adaptation of progress toward a sprint goal at least once every calendar month.

CANCELING A SPRINT



- A sprint can be canceled before the sprint timebox is over
- Only the product owner can cancel the sprint.
- A sprint would be canceled if the sprint goal becomes obsolete
- When a sprint is canceled, any completed and *done* product backlog items are reviewed
- Incomplete product backlog items are reestimated and put back on the product backlog
- Sprint cancellations consume resources since everyone regroups in another sprint planning to start another sprint

SPRINT GOAL

An objective set for the sprint that can be met through the implementation of the product backlog

- It provides guidance to the development team on why it is building the increment.
- It is created during the sprint planning meeting.
- It gives the development team some flexibility regarding the functionality implemented within the sprint.
- It can be any other coherence that causes the development team to work together rather than on separate initiatives.

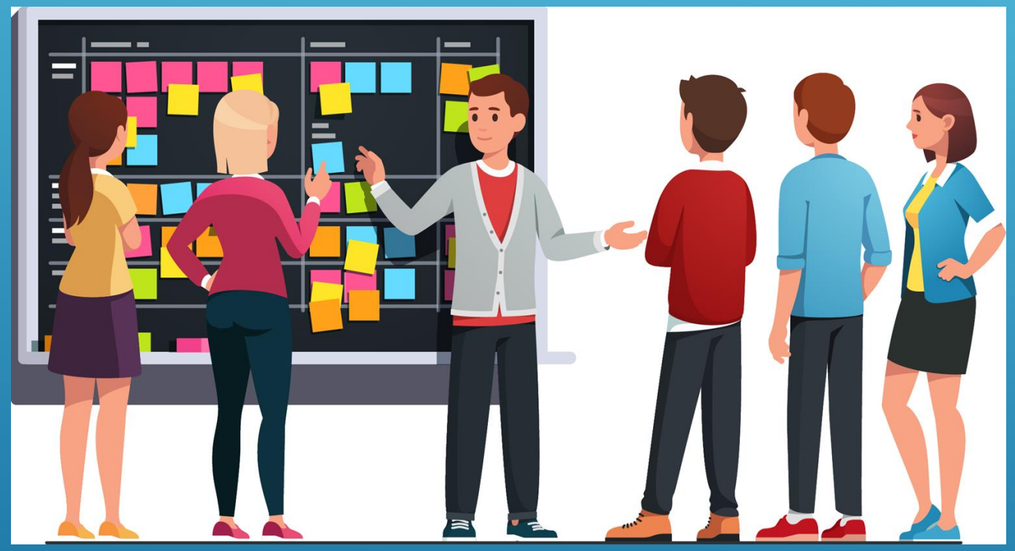
INCREMENT

An increment is the sum of all the product backlog items completed during a sprint and the value of the increments of all previous sprints.

Product Increment

- At the end of a sprint, the new increment must be *done.*
- An increment is a body of inspectable, *done* work that supports empiricism at the end of the sprint.
- It is a step toward a vision or goal.
- It must be in a useable condition regardless of whether the product owner decides to release it.

SPRINT PLANNING



- Sprint planning is where the team plans the work to be performed in the sprint.
- It is the collaborative work of the entire scrum team.
- It is timeboxed to a maximum of eight hours for a one-month sprint.
- The scrum master ensures that the event takes place and that attendants understand its purpose.
- Sprint planning answers the following:
  - What can be delivered in the increment resulting from the upcoming sprint?
  - How will be the work needed to deliver the Increment be achieved?

WHAT CAN BE DONE THIS SPRINT?

The entire scrum team collaborates on understanding the work of the sprint.

Discusses the objective that the sprint should achieve and the product backlog items that would achieve the sprint goal

Works to forecast the functionality that will be developed during the sprint

WHAT CAN BE DONE THIS SPRINT?

- Meeting inputs:
  - The product backlog
  - The latest product increment
  - The projected capacity of the development team during the sprint
  - Past performance of the development team
- Selection of product backlog items for the sprint is solely up to the development team.
- Only the development team can assess what it can accomplish over the upcoming sprint.
- During sprint planning, the scrum team also crafts a sprint goal.

HOW WILL THE CHOSEN WORK GET DONE?

- Decides how it will build the items' functionality into a *done* product increment during the sprint
- Self organizes to undertake the work in the sprint backlog, during sprint planning
- Renegotiates the selected product backlog items with the product owner if it is too much or too little work
- Invites other people to attend to provide technical or domain advice
- Explains to the product owner and scrum master how it intends to work as a self-organizing team to accomplish the sprint goal

The product backlog items for this sprint and the plan for delivering them is called the sprint backlog.

DAILY SCRUM

Daily Scrum Meeting

- The daily scrum is a 15-minute timeboxed event for the development team and is held every day of the sprint.
- It optimizes the probability that the development team will meet the sprint goal.
- Daily scrum:
  - Improves communications
  - Eliminates other meetings
  - Identifies impediments to development for removal
  - Highlights and promotes quick decision-making
  - Improves the development team’s level of knowledge

DAILY SCRUM

- Plans work for the next 24 hours to optimize team collaboration and performance
- Understands how it intends to work together as a self-organizing team
- Creates the anticipated increment by the end of the sprint
- Meets immediately after the daily scrum for any detailed discussions
- Uses questions like:
  - What did I do yesterday?
  - What will I do today?
  - Do I see any impediments?

DAILY SCRUM

- Ensures that the development team has the meeting though they are responsible for conducting the daily scrum
- Teaches the development team to keep the daily scrum within the 15-minute timebox
- Ensures people other than the development team do not disrupt the internal meeting

SPRINT REVIEW

It is held at the end of the sprint to inspect the increment and adapt the product backlog if needed.

- The scrum team and stakeholders collaborate about what was done in the sprint.
- Attendees collaborate on the next things that could be done to optimize value.
- It is an informal meeting and the presentation of the increment is intended to elicit feedback and foster collaboration.
- It is at most a four-hour meeting for one-month sprints.
- The scrum master teaches everyone involved to keep it within the timebox.

ELEMENTS OF SPRINT REVIEW

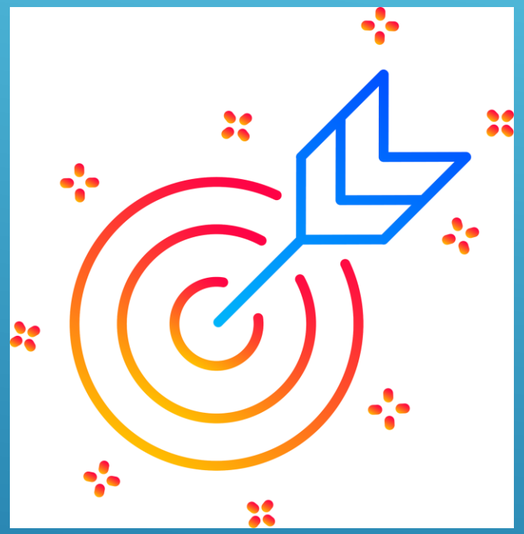
- It includes the scrum team and the key stakeholders invited by the product owner.
- The product owner explains the *done* items of the product backlog.
- The development team discusses what went well during the sprint, what problems it ran into, and how those problems were solved.
- The product owner projects likely target and delivery dates based on progress to date.
- The entire group collaborates on what to do next to provide valuable input to subsequent sprint planning.
- It includes review of how the marketplace or potential use of the product might have changed and what is the most valuable thing to do next.
- It includes review of the timeline, budget, potential capabilities, and marketplace for the next anticipated release of functionality or capability of the product.

SPRINT RETROSPECTIVE

It occurs after the sprint review and prior to the next sprint planning.

- It is a at most a three-hour meeting for one-month sprints.
- The scrum master:
  - Ensures that the event takes place and that attendees understand its purpose
  - Ensures that the meeting is positive, productive, and within the timebox
  - Participates as a peer team member in the meeting from the accountability over the scrum process
  - Encourages the scrum team to improve within the Scrum process framework, its development process, and practices

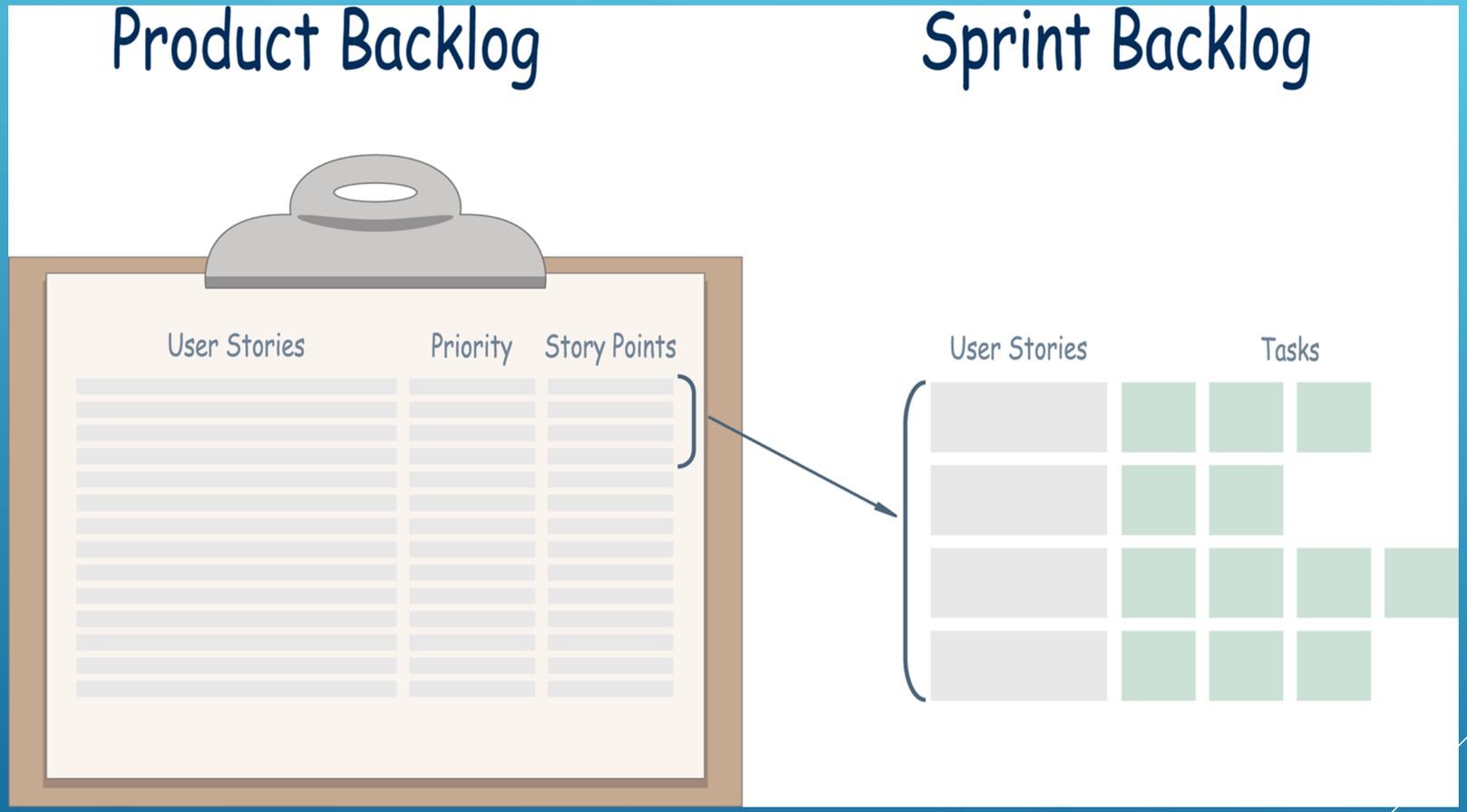
SPRINT RETROSPECTIVE



- Inspect how the last sprint went with regards to people, relationships, process, and tools
- Identify and order the major items that went well and potential improvements
- Create a plan for implementing improvements to the way the scrum team does its work

The sprint retrospective provides a formal opportunity to focus on inspection and adaptation.

PRODUCT BACKLOG



PRODUCT BACKLOG

The product backlog is an ordered list of everything that is known to be needed in the product.

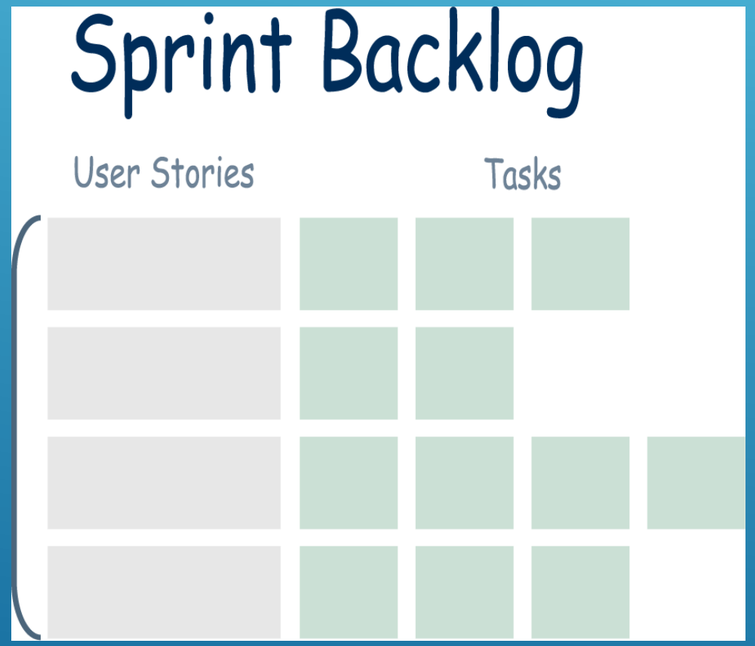
- It is the single source of requirements for any changes to be made to the product.
- It is owned by the product owner including its content, availability, and ordering.
- It constantly changes to identify what the product needs to be appropriate, competitive, and useful.
- It lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases.

PRODUCT BACKLOG REFINEMENT

- Product backlog refinement is the act of adding detail, estimates, and order to items in the product backlog.
- It is an ongoing process where the product owner and the development team collaborate on the details of product backlog items.
- The scrum team decides how and when refinement is done.
- Refinement usually consumes no more than 10% of the capacity of the development team.
- However, the product owner can update product backlog items at any time.

SPRINT BACKLOG

It is the set of product backlog items selected for the sprint and a plan for delivering the product increment and realizing the sprint goal.



- It is a forecast by the development team about what functionality will be in the next increment and the work needed to deliver that functionality into a *done* increment.
- It makes visible all of the work that the development team identifies as necessary to meet the sprint goal.
- It is a plan with enough detail that changes in progress can be understood in the daily scrum.
- It is a highly visible, real time picture of the work that the development team plans to accomplish during the sprint.

DEFINITION OF *DONE*

Definition of *done* is used to assess when work is complete on the product increment.

It guides the development team in knowing how many product backlog items it can select during a sprint planning.

DEFINITION OF *DONE*

- If the definition of *done* for an increment is part of the conventions, standards, or guidelines of the development organization, all scrum teams must follow it as a minimum.
- If *done* is not a convention, the development team must define a definition of *done* appropriately for the product.
- If there are multiple scrum teams working on a release, the development teams on all of the scrum teams must mutually define the definition of *done*.
- As scrum teams mature, it is expected that their definitions of *done* will expand to include more stringent criteria for higher quality.
- New definitions may uncover work to be done in previously *done* increments.

- PRINCE2 Agile is not only suitable for projects, but also uses a wide range of agile behaviors, concepts, frameworks, and techniques that are also used in a BAU environment.
- Agile includes situations that are large scale, complex in nature, and happening in a wide array of contexts beyond software development.
- Kanban systems are visual management systems that limit the number of work items in circulation.
- Lean Startup is a method to grow new businesses, and develop existing ones, through product innovation in uncertain markets.
- A workshop is an activity where several people come together to achieve an objective by harnessing the interactions and creativity of the participants.

1

Which of the following is NOT a principle of Agile Manifesto?

Working software is the primary measure of progress

Simplicity, the art of maximizing the amount of work not done, is essential

Avoid the use of partitioning work into timeboxes and manage work by using a queue

The best architectures, requirements, and designs emerge from self-organizing teams

1

Which of the following is NOT a principle of Agile Manifesto?

Working software is the primary measure of progress
C

Simplicity, the art of maximizing the amount of work not done, is essential

Avoid the use of partitioning work into timeboxes and manage work by using a queue

The best architectures, requirements, and designs emerge from self-organizing teams

Avoiding the use of partitioning work into timeboxes and managing work by using a queue is NOT a principle of Agile Manifesto.

2 **Which of the following is a core concept of Lean Startup that applies to PRINCE2?**

Implement feedback loops

Manage the flow

Make policies explicit

Fail fast

2 **Which of the following is a core concept of Lean Startup that applies to PRINCE2?**

Implement feedback loops
D

Manage the flow

Make policies explicit

Fail fast

Build, measure, and learn, create a minimum viable product (MVP), fail fast, and validated learning are the core concepts of Lean Startup that apply to PRINCE2.

3 **Which of the following basics of Kanban method underpins the pull system which characterizes the way Kanban avoids scheduling work at specific times?**

Visualize

The use of WIP limits

Feedback loops

Manage the flow

3 **Which of the following basics of Kanban method underpins the pull system which characterizes the way Kanban avoids scheduling work at specific times?**

Visualize

The use of WIP limits

Feedback loops

Manage the flow

B

The use of WIP limits underpins the pull system which characterizes the way Kanban avoids scheduling work at specific times, referred to as a push system, and instead pulls work from upstream.