

Introduction

- **Welcome**

- **What is this course all about?**

- Not an in-depth guide but an excellent review
 - Covers the most frequently asked concepts
 - Will help you pass the exam by focusing your study efforts

- **About the Exam**

- **Exam Description**

- First level of certification in PRINCE2 Agile
 - How to blend PRINCE2 with Agile
 - Do you know and understand PRINCE2 and Agile enough to work effectively inside of a project management team within the PRINCE2 environment?
 - Must pass Foundation before attempting the Agile Practitioner certification

- **Exam Objectives**

- Prepare for the PRINCE2 Agile® Foundation Exams
 - Understand the key aspects of PRINCE2®
 - Understand basic concepts behind common agile ways of working
 - Demonstrate the purpose of combining PRINCE2 with agile
 - Be able to fix and flex the six aspects of a project in an agile context
 - Apply the PRINCE2 principles and tailor the themes, processes and management products to a project in an agile context
 - Incorporate the areas that can support a PRINCE2 Agile implementation

- **Exam Details**

- 60 questions in 60 minutes
 - Multiple-choice (standard, list, negative)
 - Requires 28 out of 50 (56%) to pass
 - Closed Book
 - Last Updated in January 2018

- **Are You Ready?**

- Take the practice exams
 - Did you score at least 70% or higher?
 - If you need more practice, take additional practice exams to hone your skills before attempting the certification exam...

Overview of PRINCE2

- **Project Management**

- **What is Project Management?**

- Practice of initiating, planning, executing, controlling, and closing the work of a team to achieve specific goals and meet specific success criteria at the specified time

- **What is a Project?**

- A temporary organization that is created for the purpose of delivering one or more business products according to an agreed upon business case

- **What is a Business as usual?**

- Work that is conducted by an organization on a daily basis

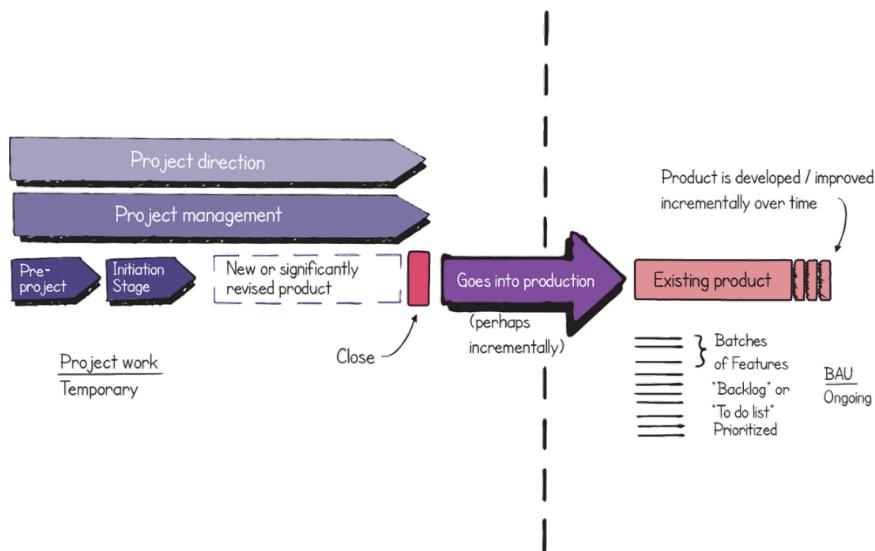
- **Project or business as usual**

- Agile can be used for both projects and business as usual
- PRINCE2® and PRINCE2 Agile are only used for projects

- **Project or business as usual**

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

- **Organizations Need Both**



- **Seven Principles**

- **Seven Principles**

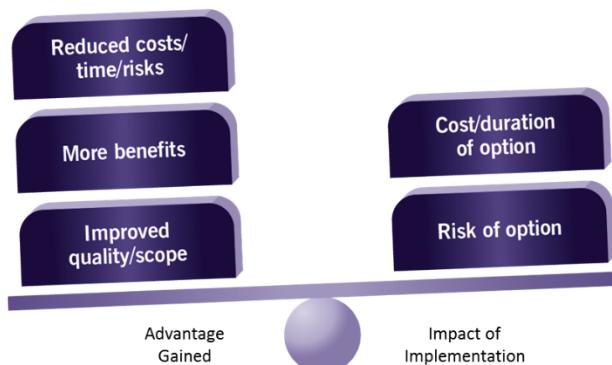
- PRINCE2 looked at numerous projects during its development and asked two questions:
 - Why did this project succeed?
 - Why did this project fail?
- The answers shaped PRINCE2 because it found the commonality within the projects...
- These best practices have been proven...
- Projects that had the 7 key principles tended to be more successful and less likely to fail...

- **PRINCE2® Principles**

- Continued business justification
- Learn from experience
- Defined roles and responsibilities
- Manage by stages
- Manage by exception
- Focus on products
- Tailor to suit the project

- **Continued Business Justification**

- Project must always remain desirable, viable, and achievable



Is there a positive balance of benefits, costs, and risks?

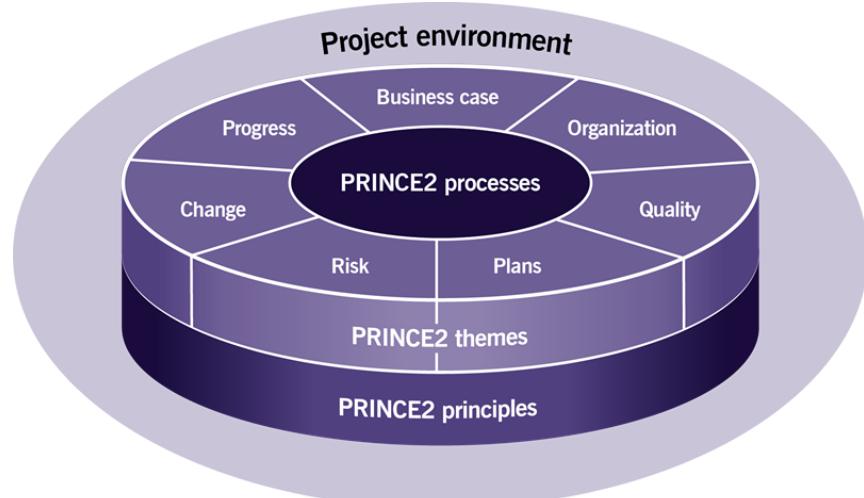
- **Learn from Experience**

- Everyone makes mistakes, but how can we learn from them?
- Collect reports of the lessons learned from previous projects
- Generate reports during the project

- **Defined Roles and Responsibilities**

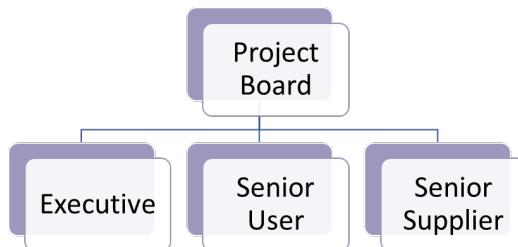
- Everyone in the project should know exactly what role they will be fulfilling and its associated responsibilities

- Provide terms of reference and role descriptions for all team members
- Ensure each team member has the correct level of authority to perform their role
- Ensure each person has ample time to dedicate to the project
- **Managed by Stages**
 - It is more accurate to plan by stages
 - Agree to the detail at each stage break
 - How much will it cost?
 - How long will it take?
- **Manage by Exception**
 - Only significant deviations from a budget or approved plan are brought to the attention of management
- **Focus on Products**
 - Successfully delivered products lead to the project's benefits being realized
- **Tailor to Suit the Project**
 - PRINCE2® is highly adaptable specifically to each and every project
 - Project occurs in different contexts
 - Culturally
 - Geographically
 - Complexity
 - Scale
 - Risk
- **Seven Themes**
 - **Seven Themes**

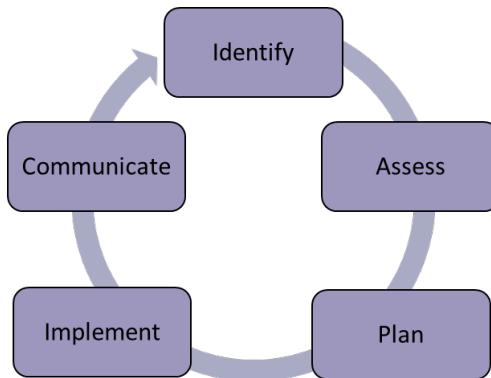


- Areas of project management that must be addressed continuously throughout the project

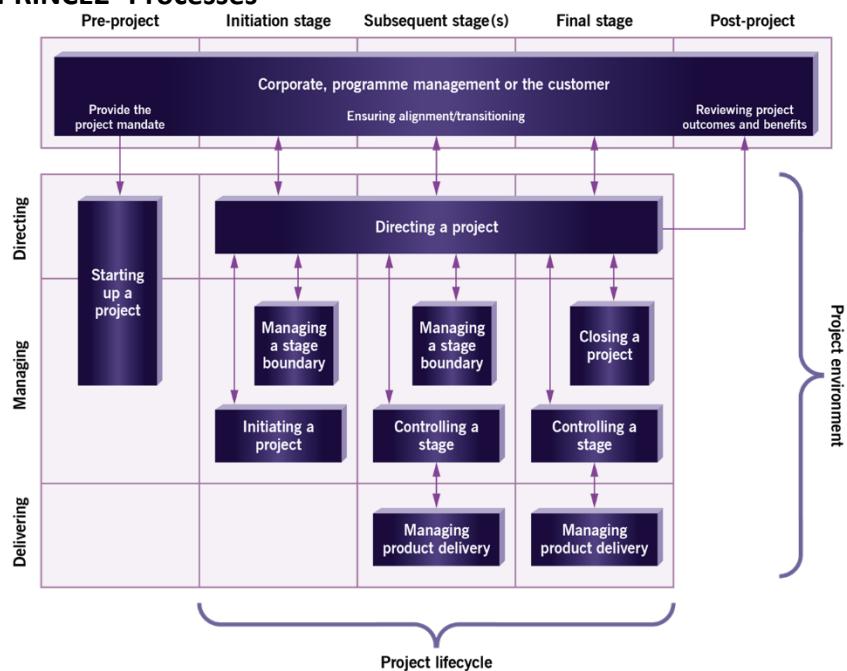
- Themes are based upon the 7 principles
- Themes are applied by using the 7 processes
- **PRINCE2® Themes**
 - Business case
 - Organization
 - Quality
 - Plans
 - Risk
 - Change
 - Progress
- **Business case**
 - Creates and maintains a business justification for the project
 - Ensures project outcomes are achieved and benefits realized
- **Organization**
 - Defines the project organization structure and roles
 - Defines the approach to communicating and engaging with stakeholders.



- **Quality**
 - Defines the project quality management approach
 - Specifies prioritized acceptance criteria for the final project product(s)
- **Plans**
 - Recommends different levels of plan to facilitate communication and control from the differing perspectives of the project organization
 - Plans enable the business case to be realized
- **Risk**
 - Defines the risk management approach and ensures that project risks are identified, assessed and controlled



- **Change**
 - Defines the change control approach and ensures that issues are captured, assessed and controlled
- **Progress**
 - Defines the way that the project progress is measured and compared to performance targets
 - Progress enables a forecast of the continuing project viability
- **7 Processes**
 - **Seven Processes**
 - Describes which role from the project management team is responsible for making decisions and when decisions should be made
 - Application of the principles and themes
 - **PRINCE2® Processes**



- Starting Up A Project
- Initiating A Project
- Directing A Project
- Managing A Stage Boundary
- Controlling A Stage
- Managing Product Delivery
- Closing A Project
- **Starting Up A Project**
 - To ensure that the prerequisites for initiating a project are in place by answering the question:
Do we have a viable and worthwhile project?
- **Initiating A Project**
 - To establish solid foundations for the project and to enable the organization to understand the work that needs to be done to deliver the project's products before committing to a significant spend
- **Directing A Project**
 - To enable the project board to be accountable for the project's success by making key decisions and exercising overall control while delegating day-to-day management of the project to the project manager
- **Managing A Stage Boundary**
 - To enable the project manager to provide the project board with sufficient information to be able to review the success of the current management stage, approve the next stage plan, review the updated project plan, confirm continued business justification, and acceptability of the risks
- **Controlling A Stage**
 - To assign work to be done, monitor such work, deal with issues, report progress to the project board, and take corrective actions to ensure that the stage remains within tolerance
- **Managing Product Delivery**
 - To control the link between the project manager and the team manager(s) by agreeing the requirements for acceptance, execution, and delivery
 - The role of the team manager(s) is to coordinate an area of work that will deliver one or more of the project's products
- **Closing A Project**
 - To provide a fixed point at which acceptance for the project product is confirmed, and to recognize that objectives set out in the original project

initiation documentation have been achieved ...or that the project has nothing more to contribute

- **Management Products**

- **Management Products**

- 26 management products that support the PRINCE2® method
 - All of them are included in Appendix A of the manual
 - All can and should be tailored, but focus on...
 - Business case
 - Checkpoint report
 - Highlight report
 - Project brief
 - Project initiation documentation
 - Project product description
 - Work package

- **Project Brief**

- Ensures the project has agreed upon and well-defined starting point

- **Project Initiation Document**

- Cornerstone document of a project that describes the plan of approach for project management
 - Contains:
 - Project plan
 - Detailed business case
 - Communication management approach
 - Risk management approach
 - Quality management approach
 - Change control approach
 - Project controls
 - Any tailoring you are performing

- **Project Product Description**

- Detail exactly what the project will deliver
 - Contains:
 - Purpose
 - Composition
 - Specifications
 - Quality criteria
 - Tolerance levels
 - Acceptance criteria

- **Business Case**

- Document that provides the justification for starting and continuing the project

- Continually referenced during the project to ensure continued business justification still exists
- **Work Package**
 - Provides a team manager with the tolerances required for product delivery
 - Approved by the project manager prior to any work beginning on the work package
- **Checkpoint Report**
 - Report that a team manager creates and sends to the project manager with the current status of their work packages
 - A time-based control used in the project
- **Highlight Report**
 - Report that the project manager creates and sends to the project board with the current status of the stage plan
 - Incorporates the checkpoint reports from all team managers into a single report
 - A time-based control used in the project

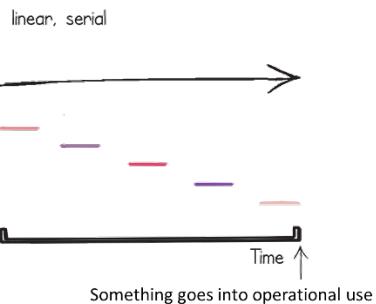
Introduction to Agile

- **What is Agile?**

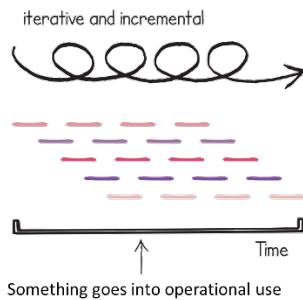
- **Overview of Agile**

- Agile addressed the new demands placed on the delivery of software
 - Term 'agile' can be viewed in many different ways
 - Several well-known frameworks are referred to as 'agile ways of working'
 - Agile is characterized by many familiar behaviours, concepts, and techniques

- **Waterfall Development**

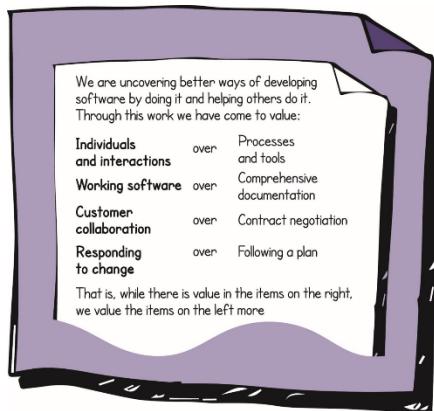


- **Agile Development**



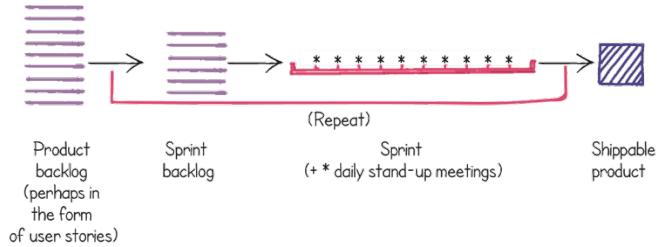
- **The Agile Manifesto**

- The closest to a single definition of agile
 - Created as an alternative to 'waterfall' processes



- **Basic Perception of Agile**

- A timeboxed approach to the development of software
- A collection of agile techniques
- Use of the Scrum framework

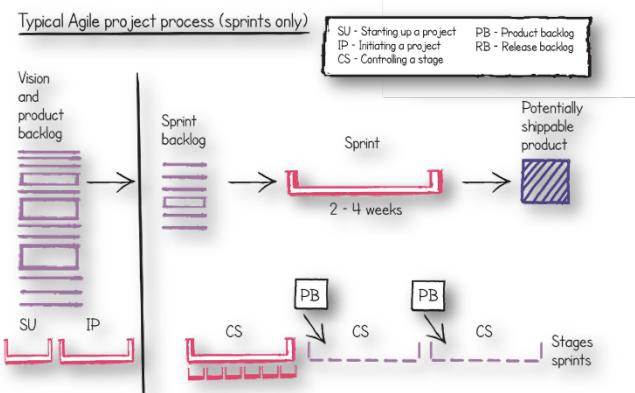


A basic Backlog and Sprint structure is commonly used with Agile

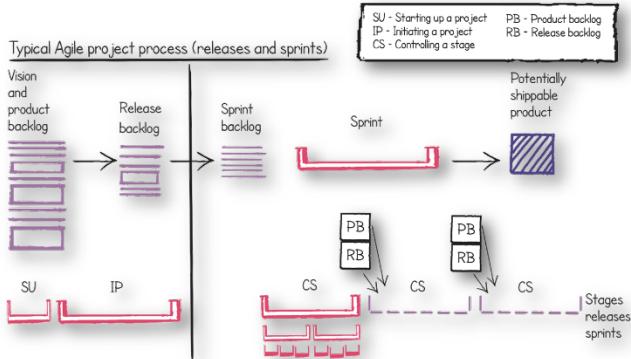
- **Beyond A Basic Perception**

- A more comprehensive view would include:
 - Vision, Roadmap, and Releases
 - Non-IT situations
 - Project work
 - Flow-based working

- **Agile Sprints Within PRINCE2 Processes**



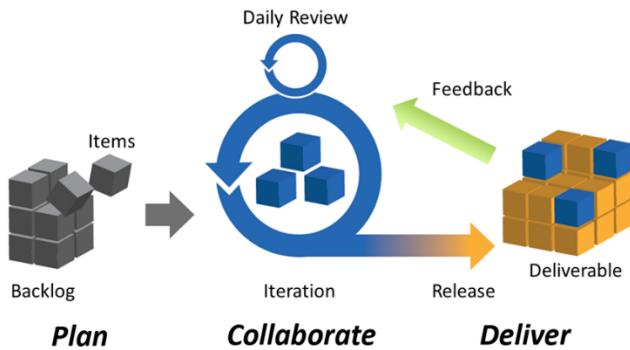
- Agile Sprints and Releases Within PRINCE2 Processes



- Agile Frameworks

- Agile Frameworks

- Many frameworks are recognized as being agile
- Some are more common than others
- Many are only in IT



- | | |
|-----------------------|----------------|
| ● <u>Scrum</u> | ● DAD |
| ● <u>Kanban</u> | ● DSDM/AgilePM |
| ● <u>Lean Startup</u> | ● DevOps |
| ● Lean | ● FDD |
| ● XP | ● Crystal |
| ● SAFe | ● ASD |

Covered in PRINCE2® Agile

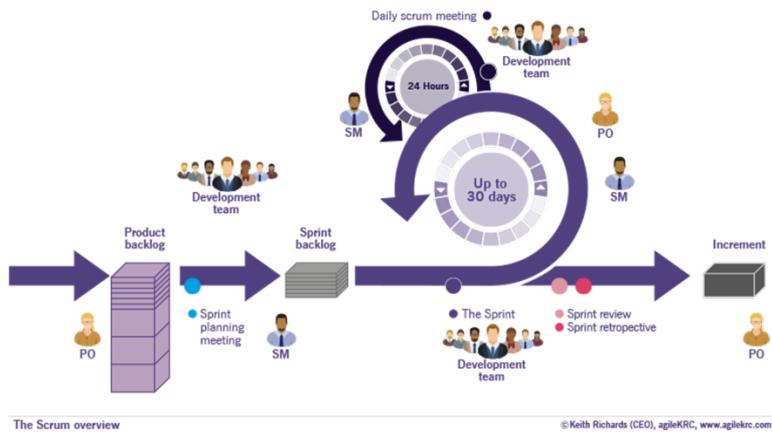
- Scrum Basics

- **What is Scrum?**
 - A framework for developing and sustaining complex products
 - A collection of roles, events, artefacts and rules
 - Created by Schwaber and Sutherland (circa 1995)
 - Method to assess the efficiency of your practices in order to improve them
- **Scrum Theory**
 - Founded on empirical process control theory
 - Decisions based on evidence
 - Three core areas:
 - **Transparency** (Significant aspects of the process must be visible to those responsible for the outcome)
 - **Inspection** (Users must frequently inspect artifacts and progress toward a Sprint Goal)
 - **Adaptation** (Processes or materials should be adjusted as needed)
- **The Scrum Team**
 - Self-organizing
 - Cross-functional
 - Flexible, creative, productive
- **Roles in The Scrum Team**
 - Product Owner
 - Responsible for maximizing the value of the product resulting from the work of the Development Team
 - Manages product backlog
 - Development Team
 - Consists of professionals who do the work to deliver a releasable Increment of “Done” product each Sprint
 - Scrum Master
 - Servant-leader who promotes and supports Scrum
- **Using Scrum**
 - **Scrum Artifacts**
 - Product Backlog
 - Sprint Backlog
 - Shippable
 - **Product Backlog**
 - Lists all the features, requirements, enhancements, and fixes proposed for future releases
 - Epics
 - Stories

- Continually refined by adding details, estimates, and order until they are “ready”
- Ready status
 - Products that can be “Done” by the Development Team within one Sprint
- **Sprint Backlog**
 - Set of items selected for the Sprint from the Product Backlog
 - Items that will create the “highest business value” are pulled by the team to work on
 - At least one high priority process improvement should be included per Sprint
- **Increment**
 - Culmination of all Product Backlog items completed during a Sprint
 - Item must be in useable condition
 - Shippable, releasable, and “Done”
- **The Five Scrum Events**
 - Sprint
 - Sprint planning
 - Daily scrum
 - Sprint review
 - Sprint retrospective
- **Sprint**
 - A time-boxed event of one-month or less
 - Creates a “Done”, useable, and releasable Increment
 - Maintains a consistent duration throughout a project
- **Sprint Planning**
 - Plan created collaboratively by the entire Scrum Team
 - Time-boxed event of less than 8 hours per month
 - What work will be delivered in this Increment?
 - How will the work needed to be performed be achieved?
- **Daily Scrum**
 - Time-boxed event of 15 minutes for Development Team
 - Daily meeting to discuss what was done in the last 24 hours and what will be done today
 - Identifies issues, risks, and challenges to finishing the Increment on time
- **Sprint Review**
 - Occurs at the end of the Sprint
 - Inspects the Increment and adapts Product Backlog
 - Informal meeting, not a status meeting
 - No longer than 4 hours in duration per month

- **Sprint Retrospective**
 - Opportunity for the Scrum Team to inspect itself
 - Creates a plan for improvements in the next Sprint
 - Occurs after Sprint Review and before Spring Planning
 - At most a 3-hour meeting per month
- **Scrum Events**
 - Events are prescribed to create regularity
 - Every event has a maximum duration
 - The Sprint is at the heart of this concept
 - Forces transparency
 - Opportunities to inspect and adapt

- **Putting It All Together**

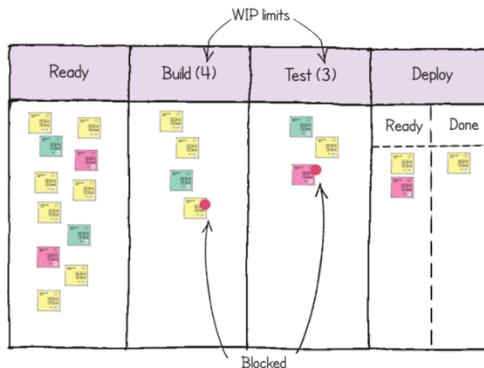


- **Kanban**
 - **The Kanban Method**
 - Visual management systems that limit the number of work items in circulation
 - Kanban increases agility through:
 - Improved day-to-day decision making
 - Deferral of commitment
 - Reduced lead times
 - Applicable in a project context to time-boxes in PRINCE2 Agile
 - **6 General Practices of Kanban**
 - Visualization
 - Limiting 'Work in Progress' (WIP)
 - Manage the flow

- Making policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

- **1. Visualization**

- Show how work is progressing, what is left to do, and what problems exist



- **2. Limiting 'Work in Progress'**

- Fundamental concept in Kanban
 - May appear counterintuitive at first
- WIP limits underpin the 'pull' system
- Kanban avoids scheduling work at specific times
- It pulls work when capacity exists
- Reduces the impact of task switching and multitasking

- **3. Manage the Flow**

- Team constantly looks at ways to maximise flow
- Waste is removed as quickly as possible

- **4. Making Policies Explicit**

- Boundaries clearly defined for how a team works
- Policies should evolve over time

- **5. Implement Feedback Loops**

- Value being delivered is ultimately judged by the final consumer
- Quantitatively assessing this will directly affect what will subsequently be delivered

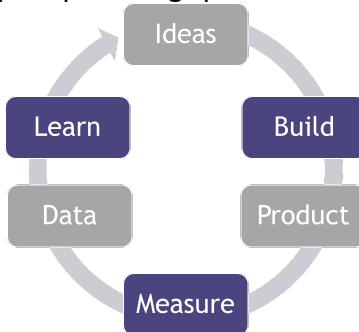
- **6. Improve Collaboratively, Evolve Experimentally**

- Method builds on collaboration through experimental improvement
- Process improvement is everyone's business every day

- **Other Considerations**

- Scrumban is the application of Kanban where the underlying process is based on Scrum
- Policies may exist for similar work items as flow may be more predictable

- Team looks to improve how the system works by carrying out experiments in a controlled and objective way
- **Lean Startup**
 - **The Lean Startup Method**
 - Methodology for developing business and products
 - Shortens development cycles by conducting:
 - Experimentation
 - Iterative product releases
 - “Fail fast, learn fast”
 - Validated learning
 - **Minimum Viable product (MVP)**
 - Version of a product that allows the team to collect the maximum amount of validated learning with least effort
 - Tests fundamental business hypotheses known as “leap-of-faith assumptions”
 - Allows companies to determine quickly if an idea is really viable and worth continued investment
 - An MVP may not go into operational use and may only be considered an experiment
 - **Continuous Development**
 - Heavily used in software development
 - All code is written for an application and immediately deployed into production
 - Reduces development cycles
 - Can lead to daily production releases
 - **Split Testing (A/B Testing)**
 - Different versions of a products are created and offered
 - Goal is to observe differences in behaviour between customer groups
 - A/B Testing should occur at the same time to minimize variables during the test
 - **Build-Measure-Learn**
 - Process loop emphasizing speed in development





PRINCE2 Agile Foundation (Study Notes)

Blending PRINCE2 and Agile

- **PRINCE2® and Agile**

- **The PRINCE2® Agile View**



- **PRINCE2® Journey with Agile**

- How PRINCE2 may look in an agile context
- This is just '**a way**' not '**the way**'
- Tailoring PRINCE2 depends on the project context, which may affect:
 - the level of formality
 - where the emphasis is placed
 - how it is carried out



- **Agile Behaviours, Concepts, And Techniques**

- A variety of behaviours, concepts and techniques exist that are seen as being part of the Agile way of working in addition to the different agile frameworks

- **Agile Behaviours**

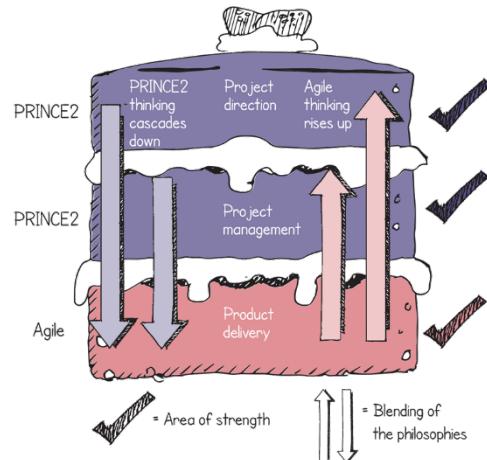
- Also called Principles, Values, or Mindset
- Examples
 - Being collaborative
 - Self-organizing
 - Customer-focused
 - Empowered
 - Trusting not blaming

- **Agile Behaviours in PRINCE2**

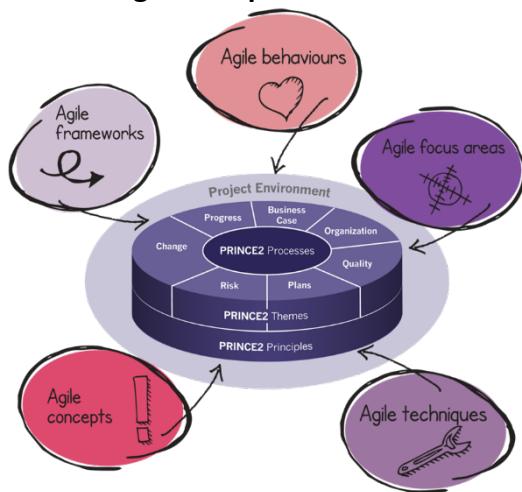
- Transparency
 - Openness, visibility, honesty, trust, integrity and respect

- Collaboration
 - Internal (the team work together) and external (engaging with customers) leads to shared understanding and ownership
- Rich communication
 - Face to face in preference to words alone
- Self-organization
 - Trust the people closest to the work to know best
- Exploration
 - Frequent iterations and rapid feedback loops provide an opportunity to learn (experiments and spikes)
- **Agile Concepts**
 - Also called Fundamentals
 - Examples
 - Prioritizing what is delivered
 - Working iteratively and incrementally
 - Not delivering everything
 - Time-focused
 - Inspect and adapt
 - Kaizen (Japanese for “continuous improvement”)
 - Limiting work in progress (WIP)
- **Agile Techniques**
 - Also known as Practices and Tools
 - Examples
 - Burn charts
 - User stories
 - Retrospectives
 - Timeboxing
 - Measuring flow
- **Blending PRINCE2 And Agile Together**

- PRINCE2 and Agile each have their own strengths
- PRINCE2 focuses on direction and management
- Agile is delivery-focused



- **What Does PRINCE2 Agile Comprise of?**



- **Eight Guidance Points**

- PRINCE2 is already enabled for use with Agile
- PRINCE2 is suitable for any style of project and is not a 'traditional' project management approach
- PRINCE2 Agile is for use with any project (not just IT)
- IT-only frameworks and techniques are not the focus
- Much more to Agile than Scrum (Agile is not Scrum)
- Scrum and Kanban are most commonly used approaches but not suitable for managing a project by themselves
- Agile is a family of behaviors, concepts, frameworks, and techniques
- Using Agile in a project is a question of 'how much'

- **Agile Doesn't Mean Removing Control and Governance**

- Control and governance allow Agile to be used in complex environments

- **Fix and Flex**

- **PRINCE2 Performance Variables**

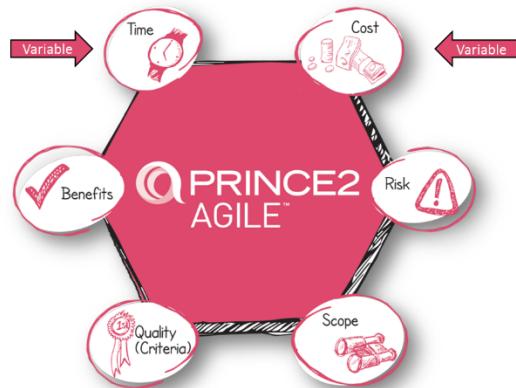


- **Tolerance**

- An allowable variation around targets
 - An exception should be raised if tolerances are exceeded

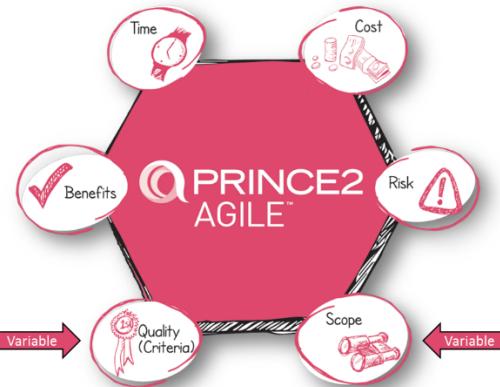
- **Performance Variables in Waterfall Development**

- Time and cost are often seen to be the most significant variables



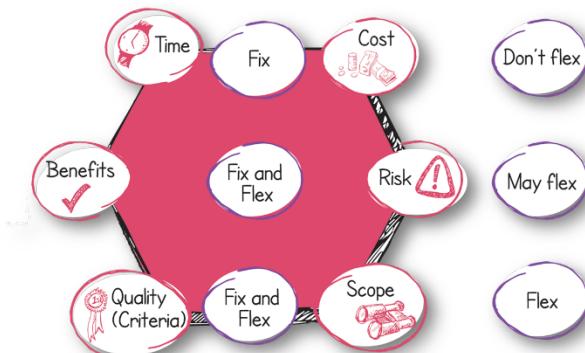
- **Performance Variables in Agile Development**

- Scope and quality are often seen to be the most significant variables



- **The Hexagon**

- Tolerances for the different objectives can be fixed or flexed



- **Time (Fix)**

- Zero tolerance for extra time on all levels of a plan
- Agile uses time-boxing with set durations

- **Cost (Fix)**

- Zero tolerance for extra cost on all levels of a plan
- Agile uses fixed cost and any tolerance is an exception

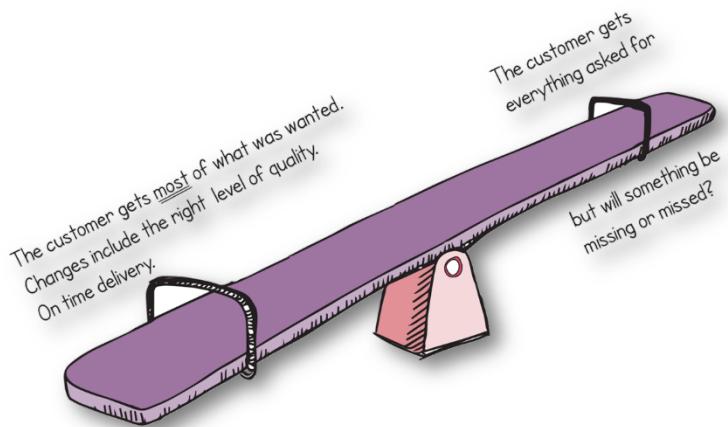
- **Quality (Fix and flex)**

- Acceptance criteria and quality criteria may be prioritized because they are not all equally important
- Essential items
 - Zero tolerance on quality expectations and acceptance criteria
- Desirable but not essential items
 - Tolerance may be utilized

- **Scope (Fix and flex)**

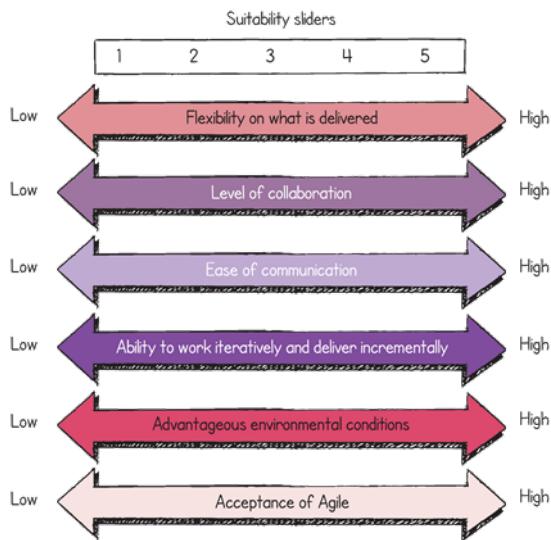
- Prioritize scope since not all items are equally important
- Essential items
 - Zero tolerance

- Desirable but not essential items
 - Tolerance may be utilized
- **Risk (Fix or flex)**
 - Defined according to the needs of the project board and project manager based on the situation
- **Benefit (Fix or flex)**
 - Zero tolerance up to the level of 'minimum viability' as defined in the business case
 - Tolerance is allowed above the 'minimum viability' level
- **The Appropriate Balance**



- **Five Targets**
 - **The Five Targets**
 - Five targets represent the **reasoning** and **why** behind the hexagon
 - Be on time and hit deadlines
 - Protect the level of quality
 - Embrace change
 - Keep teams stable
 - Accept that the customer doesn't need everything
 - **1. Be on Time and Hit Deadlines**
 - Early realization of benefits
 - Helps with planning
 - Gives confidence
 - There may be no choice
 - Reduces the likelihood of cost overruns
 - Improves reputation
 - **2. Embrace Change**

- It is inevitable
 - More accurate final product is more likely to be produced
 - Can be handled by flexing what is delivered
- **3. Protect the Level of Quality**
 - To ensure that the appropriate level of quality is achieved in order to achieve the desired outcomes
 - Poor quality can occur with...
 - Reduced testing
 - Incomplete documentation
 - Sub-optimal design
 - Lack of appropriate training
 - Non-compliance with standards
 - **4. Keep Teams Stable**
 - Agile utilizes self-organizing teams and informal communication
 - Changing team members (in the short term) can have a detrimental effect:
 - Time spent bringing new team members up to speed
 - Number of communication lines in the team grows exponentially
 - Opportunity cost incurred by the areas providing the new people
 - Team dynamics change and need to be re-established
 - **5. Accept That the Customer Does Not Need Everything**
 - Features of a product are the safest and most sensible area to compromise on when necessary
 - Not everything defined at the start must be delivered
 - Many functions and features are rarely or never used
 - Removing features helps when trying to hit deadlines and protect the level of quality for essential items
 - Delivers what the customer really wants more quickly
- **Agile Focus Areas**
 - **Agiometer**
 - A way of assessing the agile environment in order to tailor PRINCE2 in the most effective way
 - Shows risk areas with using agile
 - Shows benefits of using agile
 - Guide to make an informed decision
 - Performed pre-project and repeated at stage boundaries
 - The project manager facilitates the assessment
 - **The Six Areas Assessed**

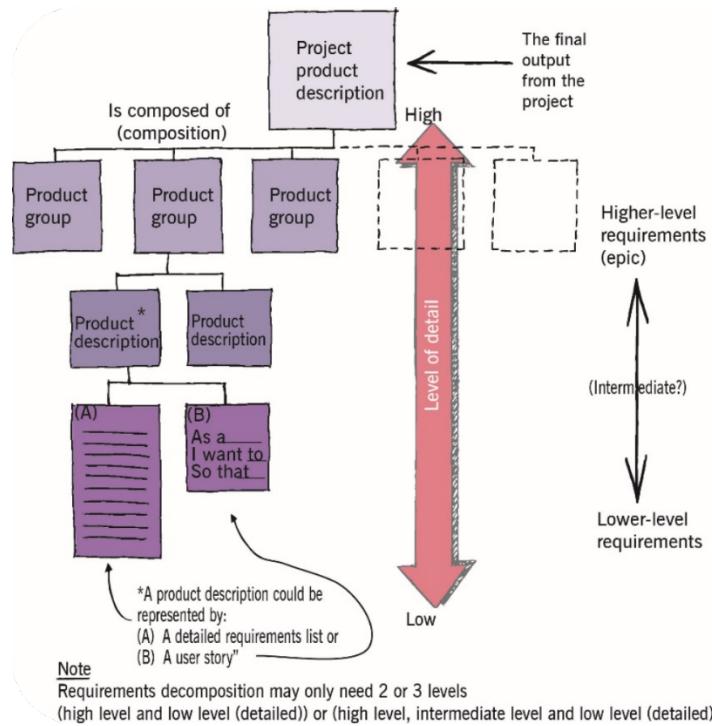


- **Using the Agilometer**

- ‘How much’ and not ‘yes or no’
- Use the sliders in isolation
- Do not add them up or create an average
- Represents a starting point but it can change
- After the assessment can any sliders be improved?

- **Requirements**

- Requirements represent the currency of an agile project
- Higher level (*MoSCoW prioritization*)
 - Project product description
 - Vision
 - Product groups
 - Epics
 - Features
 - High-level requirements
- Lower level (*Prioritized by order*)
 - Project descriptions
 - Requirements
 - Features
 - User stories
 - Features
 - High-level requirements



- **Rich communication**
 - Communication problems must be proactively addressed
 - Effective communication is fundamental to agile
 - Communication takes place in many formats
 - Communication channels
 - Written word
 - Visualization
 - Verbally by telephone
 - Verbally with face-to-face
- **Rich communication tactics**
 - Face-to-face is the fastest and clearest channel
 - Technology and level of formality needs to be considered
 - Written word has its place but it also has many disadvantages
 - Project manager (or team manager) needs to be aware of how a team is communicating
 - Communication needs to be organized and planned.
- **How Should I Communicate?**



- **Frequent Releases**
 - Release something usable (prototype or beta)
 - Requires planning
 - Product-based Planning can be used
 - Advantages
 - Early delivery of benefits to the customer
 - Allows for feedback
 - Reduces risk of delivering the wrong product
 - Gives confidence of how the project is progressing
 - Fosters stakeholder engagement
 - Makes releasing easier
- **Retrospectives**
 - Significant technique when working in an agile way
 - Reviews the way of working (processes not products)
 - Must be planned and structured
 - Cover what went well and what didn't go so well
 - Improve little by little, and little and often
 - Keep them effective by introducing variety
 - Feedback can come in the form of facts or feelings
- **Workshops**
 - Run by a neutral facilitator who managed the workshop
 - Harnesses interactions and creativity
 - Used at any point during a project
 - Can create high quality outputs quickly if you conduct the proper preparation
- **Running Effective Workshops**
 - Preparation is essential

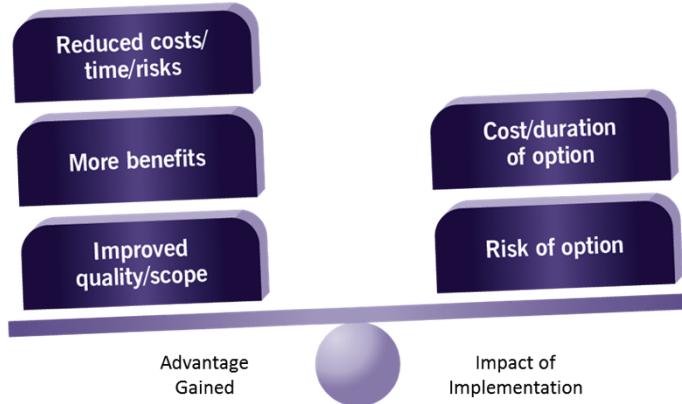
- Workshop objective
- Attendees
- Agenda
- Logistics
- Pre-reading
- Useful techniques
 - White Boards
 - Group working
 - Sticky notes

The Seven Principle, Seven Themes and Agile

- **7 Principles and Agile**

- **Continued Business Justification**

- Project must always remain desirable, viable, and achievable



How to implement in Agile: Value and Minimum Viable Product

- **Learn from Experience**

- Everyone makes mistakes, but how can we learn from them?
- Implemented through...
 - Retrospectives
 - Short feedback loops
 - Inspect and adapt

- **Defined Roles and Responsibilities**

- Everyone in the project should know exactly what role they will be fulfilling and its associated responsibilities
- Implemented through...
 - Blending PRINCE2 roles and additional agile roles

- **Manage by Stages**

- It is more accurate to plan by stages
- Agree to the detail at each stage break
- How much will it cost?
- How long will it take?
- Implemented through...
 - Timeboxes (Releases and Sprints)
 - Shorter stages to support innovation

- **Manage by Exception**

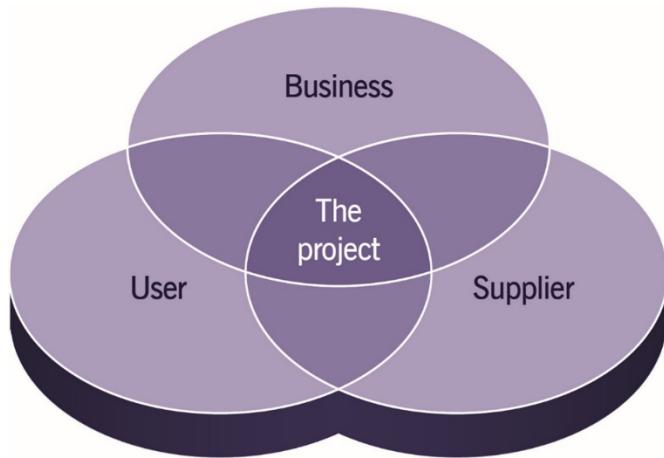
- Only significant deviations from a budget or approved plan are brought to the attention of management
- Implemented through...

- Tolerances to empower your team members
- **Focus on Products**
 - Successfully delivered products lead to the project's benefits being realized
 - Implemented through...
 - Prioritization of products and quality criteria
 - Changing format or formality of management products
- 
- **Tailor to Suit the Project**
 - PRINCE2® is highly adaptable specifically to each and every project
 - Project occurs in different contexts
 - Culturally
 - Geographically
 - Complexity
 - Scale
 - Risk
 - Implemented through...
 - Agile assessments with the Agilometer
- **7 Themes and Agile**
 - **Seven Themes**
 - Areas of project management that must be addressed continuously throughout the project
 - Themes are based upon the 7 principles
 - Themes are applied by using the 7 processes
 - **PRINCE2® Themes**
 - Business case
 - Organization
 - Quality
 - Plans
 - Risk
 - Change
 - Progress
- **Business Case**
 - **Defining Value in The Business Case**
 - Value
 - Benefits delivered in proportion to resources put in
 - Can often be subjective

- Easier to assign relative values
 - Value can be assessed in many ways
 - E.g. revenue, reducing costs, customer satisfaction
 - Focusing on benefits is helped through collaboration
- 

```

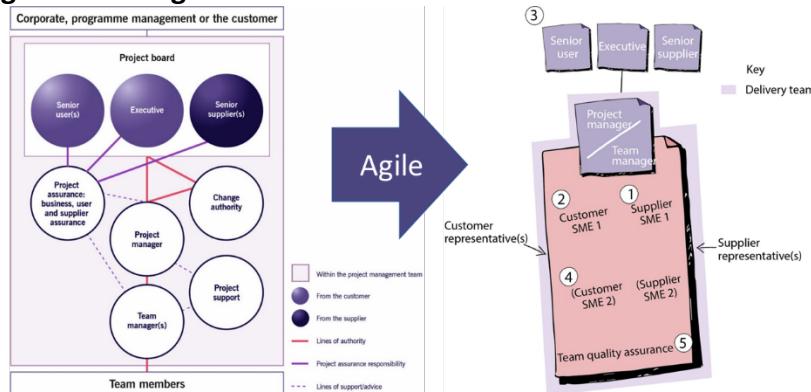
    graph LR
      A[OUTPUT] --> B[OUTCOME]
      B --> C[BENEFIT]
  
```
- **Tailoring Themes: Business Case**
 - No major changes are required
 - More info on tolerances around benefits
 - Implications of incremental delivery
 - Early benefit and early costs
 - Agile focuses on value accrued from a story
 - Allows for better feature prioritization when needed
 - **Agile View: Business Case**
 - May not exist in some agile environments
 - May be a focus on value assigned to features instead
 - May be created at the beginning as part of 'sprint zero'
 - **Guidance: Business Case**
 - May be affected by the amount being delivered
 - Early delivery of benefits will affect the business case
 - Minimum viable product (MVP) needs to be clearly stated
 - Project viability is not the same as the MVP
 - Best-case, worst-case, and expected-case will relate to the amount of features to deliver
 - High uncertainty leads to a more minimal business case
 - Plan to test assumptions quickly
 - **Organization**
 - **The Three PRINCE2 Project Interests**



- **Tailoring Themes: Organization**
 - No changes required but additional delivery roles may be needed
 - Consideration needs to be given to:
 - The team manager role
 - How it might be integrated into the delivery team
 - Common agile roles
 - E.g. product owner, scrum master, agile coach, business ambassador
 - The senior user role
 - Acting as a super product owner
 - The scrum master
 - Liaison with the project manager
 - Management by exception to enable self-organization
- **Agile View: Organization**
 - Focus on self-organizing teams
 - Two common roles:
 - Scrum Master
 - Product Owner
 - Less prominence for:
 - Project Manager/Team Manager role
 - Requirements Engineer/Business Analyst (or similar) role
 - PRINCE2 adjustments
 - Relationship between the PM and the team
 - Servant Leadership
 - Integrating the Team Manager
 - Senior User acting as a product manager
 - Defined in the project initiation documentation
 - Agile adjustments
 - Usually a single Product Owner

- PRINCE2 Agile recommends further roles added
 - Subject matter expert
 - Representative
- Defined in the working agreements
- **The Delivery Team**
 - Represented by some (or all)
 - Someone to lead the team
 - Someone from the customer (or their representative)
 - Team to create the product
 - Someone to assure the quality of the product
 - Someone to coach the team (including how to use agile)
 - Multi-skilled ‘generalizing specialists’

- **Agile View: Organization**



Working agreements can help to document roles and responsibilities

- **Guidance: Organization**

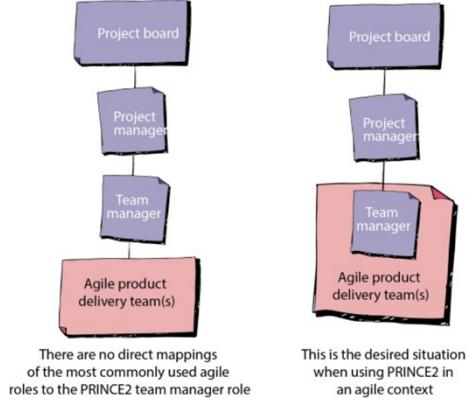
- Adding PRINCE2 roles to agile delivery roles is straightforward
- How easy it is depends upon the nature of the work
- Roles need to be aligned

Type of work	Size/scale of work	Typical number of teams involved	Ease of synchronization
Routine	Small	One	Not applicable. Handled as BAU.
Difficult	Small	One	Probably straightforward. The project manager and team manager roles are combined. The most common agile roles should be aligned easily.
Difficult	Large	Many (at least more than one)	Probably requires a degree of care. The most commonly used agile roles need to be carefully aligned as there is more than one team and the alignment may not be obvious.

- **Integrating the Project manager into Delivery Team**

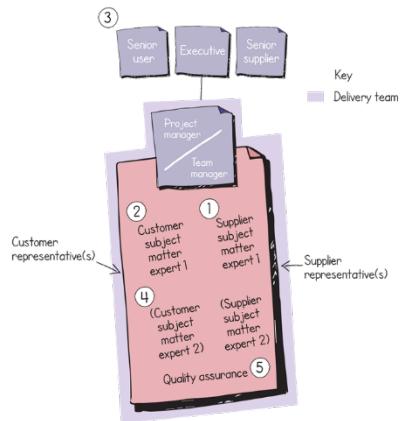
- Three options
 - Leave the delivery team roles as they are
 - Leave the delivery team roles as they are but identify a single point of contact for the project manager

- Create a team manager role in the delivery team
- Choice will be made according to the project's circumstances



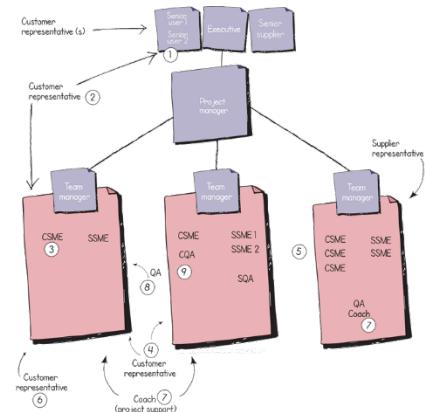
○ Single-Team Structure

All roles need to be conversant with working in an agile way



○ Multiple-Team Structure

All roles need to be conversant with working in an agile way



● Quality

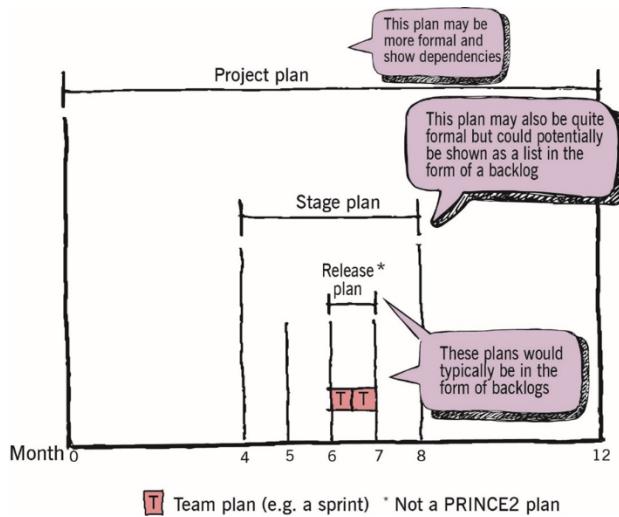
- **Tailoring Themes: Quality**
 - Protect the “fit for purpose” of the products
 - Prioritizing acceptance criteria and quality criteria
 - Defining quality tolerances
 - Differentiating between functional and non-functional requirements
 - Use agile concepts to help clarify quality criteria
 - Definitions of Ready and Done
 - Consider the frequency of quality checking and the impact it has on the way the project is planned and run
- **Quality in Relation to Scope**
 - A reduction in scope is not seen by PRINCE2 as a reduction in quality
 - Customer quality expectations and acceptance criteria are set and this level needs to be maintained



- **Agile View: Quality**
 - May not be a lot of emphasis given to quality planning and quality management during the start of a project
 - Prominent techniques such as the definition of ‘Done’ and acceptance criteria address quality control
 - Evolving the definition of ‘Done’ is commonly used
 - Concepts such as ‘test as you go’ or ‘test first’ are used for testing and quality checking
- **Guidance: Quality**
 - Product descriptions are flexible
 - User stories are a good example of this
 - Project product description’s purpose would preferably be defined as an outcome
 - Quality management and quality planning includes...
 - What tools and approaches are to be used
 - Role of the customer (an essential ingredient)

- Assessing and costing the resources
- Quality control considerations
- **How Do You Test Quality?**
 - Care needed when transferring these concepts from the software development domain
 - Common agile terms
 - Test-driven development (TDD)
 - Behaviour-driven development (BDD)
 - Definition of ‘Done’
 - Definition of ‘Ready’
 - Refactoring
 - Technical debt
- **Test-Driven Development (TDD)**
 - Concept of writing tests or quality checks before building the product or sub-products
- **Behaviour-Driven Development**
 - More collaborative than test-driven development
 - Quality checks are written using language the customer can best understand
 - What does the customer want to achieve?
- **‘Done’ Definition**
 - Set of criteria that is used to determine if a piece of work or a collection of work items is completed
 - Something is either ‘done’ or ‘not done’
- **‘Ready’ Definition**
 - Set of criteria that is used to determine if a piece of work or a collection of work is ready to be started
- **Refactoring**
 - Process of changing a software system in such a way that it does not alter the external behaviour of the code yet improves its internal structure
 - Commonly used in software developments but can be applied to any product
- **Technical Debt**
 - Refers to the eventual consequence of poor system design or development
 - Refers to work that should have been completed prior to moving on to the next step, but hasn’t
 - Not addressing the technical debt can lead to further disorder and risk to the project
- **Plans**

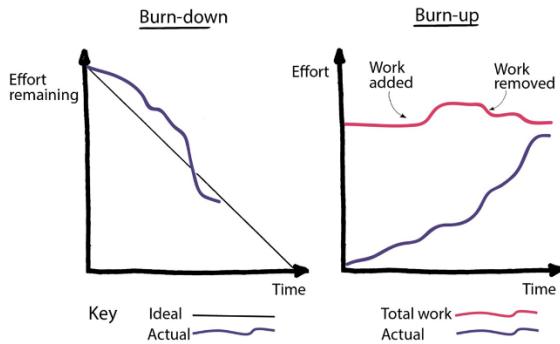
- **Tailoring Themes: Plans**
 - No changes are required
 - Many Agile techniques/approaches exist
 - Often informal and low tech (sprint planning, backlog list, ...)
 - Looks at how much (or how much value) can be delivered in a fixed timeframe
 - Releases, sprints, burn charts
 - Higher level plans may use Gantt charts and formal milestones to demonstrate the duration of work
 - Synchronize high level plans and low-level backlogs
- **Agile View: Plans**
 - A lot of emphasis is put on planning
 - Mostly relies on Empiricism instead of Rationalism
 - Planning style
 - Based on features
 - Team-based exercise
 - Plan at the last responsible moment
 - Using a point scoring system to estimate is popular
 - PRINCE2 and Agile consider the planning horizon
- **Guidance: Plans**
 - PRINCE2 supports any planning style
 - Product-based planning can be used for any plan level
 - High level plans support continued business justification
 - Provide broad timings (end date)
 - Provide broad costs (significant expenditure)
 - Lower level plans support delivery
 - Provide ready stories
 - Provide clear priorities
- **Levels of Plan**
 - Project level
 - Likely to be more formal
 - Stage boundaries, releases, major products, major activities, groupings, and dependencies
 - Broad costs and timings
 - Team level and/or release plan
 - More likely to be a backlog
 - Stories
 - Priorities
 - Amount of work
 - Relative size of work



- **Agile Estimating**
 - Uses relative estimates based upon system of points
 - Popular Types
 - Numbers based upon the Fibonacci sequence
 - 1,1,2,3,5,8,13,21,...
 - T-shirt sizes
 - S,M, L, XL, XXL
 - Story Points
 - Velocity is based upon actual delivery
 - Used to forecast future rates of progress
- **Risk**
 - **Agile Working Can Reduce Risk**
 - Agile techniques address the familiar project risks by...
 - Avoiding too much detail at the start
 - Daily stand-ups
 - Frequent delivery of product
 - Frequent demos
 - Customer interaction
 - Self-managed teams
 - **Tailoring Themes: Risk**
 - Agile working has its own set of potential risks
 - Example: challenges with continual customer engagement
 - Risk management processes don't have to be bureaucratic
 - Formality level should be project appropriate
 - Using a few columns on the team board instead of an electronic risk register
 - **Considering Risk**

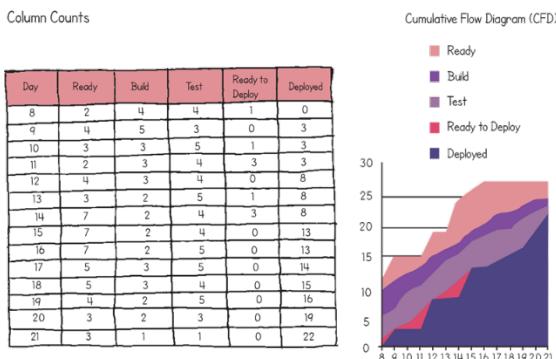
- Agile places less prominence to the area of risk
 - Agile concepts mitigate many risks associated with other approaches (e.g. waterfall)
 - PRINCE2 brings a level of formality and planning to risk management, but PRINCE2 Agile allows for less formality
 - Address risk during stand-up meetings
 - Agile has risk areas of its own (assessed by The Agilometer)
 - The five behaviours in PRINCE2 Agile helps to manage risk
- **Change**
 - **Tailoring Themes: Change**
 - Change is inevitable in PRINCE2 and Agile
 - Must control significant change
 - Level where the project was justified
 - Enable responsive change at the detail level
 - Level where change improves the quality and usability product
 - Product descriptions (quality criteria and tolerance) and work packages provide...
 - Clear baselines that can be managed formally
 - Escalate changes to the project board or to a change authority
 - Detail level change within defined tolerances that can be managed by the team dynamically
 - Requirement definitions
 - Can be binary (y/n) or a spectrum (scale)
 - Using good risk management and configuration management can lessen the impact of change
 - **Guidance: Change**
 - PRINCE2 could be said to be more cautious
 - Blending with agile controls significant change while allowing responsive change at the detail level
 - Gives us the best of both worlds...
 - Baselines must use the correct level of detail for the blending to work properly
 - Created during Starting Up a Project and Initiating a Project
 - **Change Is Empowered by The Feedback Loop**
 - Goal
 - Gather feedback as quickly as possible
 - We desire ‘true’ feedback from the end customer
 - Many forms of the feedback loop exist
 - OODA

- Observe, Orient, Decide, Act
- PDCA
 - Plan, Do, Check, Act
- PDSA
 - Plan, Do, Study, Act
- Build, Measure, Learn (Lean Startup)
- **Progress**
 - **Tailoring Themes: Progress**
 - No changes required
 - Agile techniques and approaches
 - Tracking what is delivered (velocity, lead times, or value)
 - Burn down and burn up charts
 - Demonstrating value accrued
 - Tolerances would be set to support this (scope and quality)
 - Progress is tracked at all levels of the project
 - Agile techniques and PRINCE2 processes both have value in measuring progress
 - **Agile View: Progress**
 - Agile puts a large emphasis on tracking progress
 - Prefer visualization techniques
 - Burn charts
 - Work in Progress (WIP) boards
 - Audience for any technique will need to be comfortable with the format
 - Transparency is key to visualizing and communicating progress
 - **Burn Charts**
 - Popular technique displayed as graph
 - Burn-down chart
 - Burn-up chart
 - Shows the current situation, rate of progress, and what has been completed
 - Burn-down charts assume a fixed amount of work
 - Burn-up charts should be used if the amount of work is likely to change
 - **Burn-Down and Burn-Up**



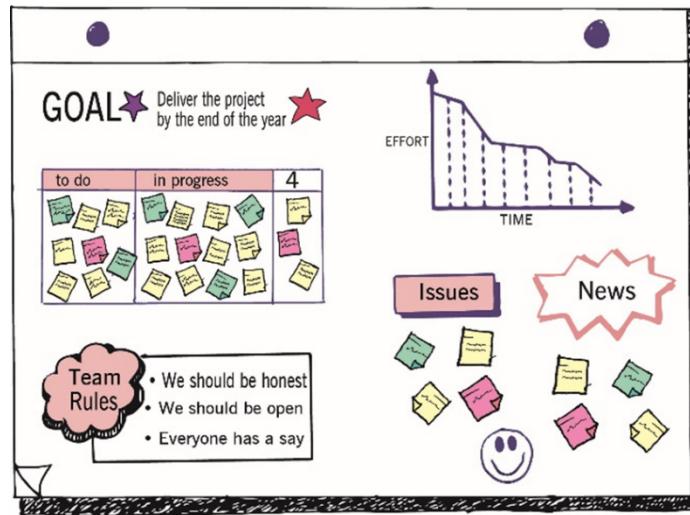
- **Cumulative Flow Diagrams (CFDs)**

- Track work items and shows the amount of work in each column each day
- Work in Progress (WIP) is the vertical difference between the top and bottom lines whereas the horizontal difference shows the lead time



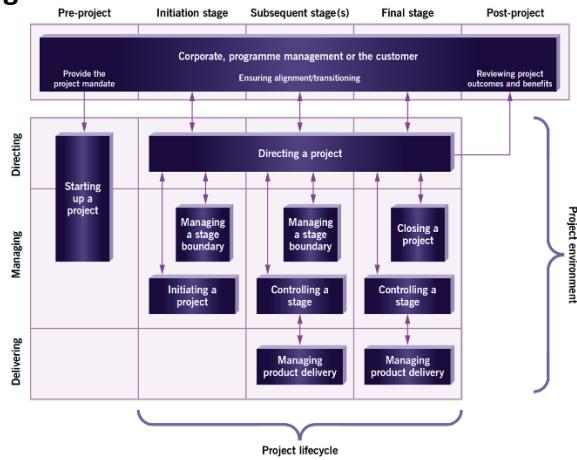
- **Information Radiators**

- Visual information that's immediately accessible and contributes to transparency
- Can display a wide variety of information
- Best created and maintained manually
- Information is 'pushed' as opposed to 'pulled'
- Holding a daily stand-up meeting by a display enables it to be updated immediately



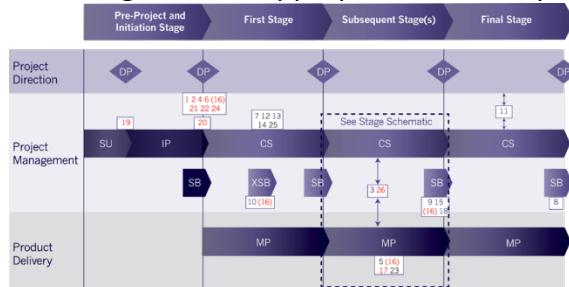
The Seven Processes and Agile

- **7 Processes and Agile**

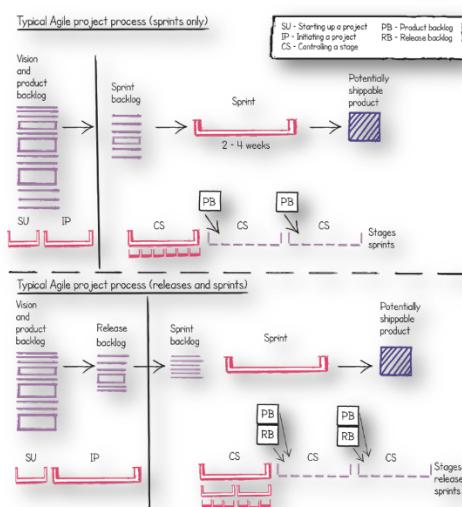


- **Agile and the PRINCE2 Processes**

- Agile needs to be incorporated into all seven processes
- Amount of agile that is appropriate for each process varies



- **Blending Agile and PRINCE2**

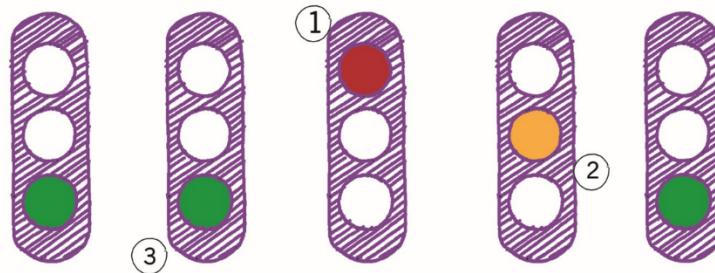


- **Starting Up A Project and Initiating A Project**

- **Tailoring Starting Up A Project and Initiating A Project**
 - Determine if the project is viable and worthwhile
 - Create a solid foundation to understand the work
 - Set the project up in an appropriate manner
 - Integrating with agile teams (e.g. role names)
 - Impact of frequent releases of products to enable and provide benefits
 - Define things at the right level
 - Outputs and Outcomes
 - Project Product Description
 - Benefits
 - Business case (Best and worst amount of product)
 - High level requirements (epics)
 - Define things in the right way
 - To enable agile to work easier (e.g. outcome focussed)
- **Project Brief**
 - Likely to be informal
 - Project definition is more outcome based
 - Impact of frequent delivery considered
 - Lean startup and MVP
 - Includes the project approach that will discuss the use and benefit of Agile working
- **Business Case**
 - Impact of flexing amount delivered considered
 - MVP identified
 - Best case and worst case described in terms of amount delivered
- **Project Product Description**
 - Focus on outcome desired
 - Created as part of a workshop
 - Composition (major products) might be similar to epics
 - Creation of the product backlog
- **Project Initiation Document**
 - Enough and no more
 - May exist as an information radiator
 - Plan the frequency of releases
 - Write a definition of done
 - Map PRINCE2 and agile roles
 - Describe the tailoring undertaken
- **Starting Up A Project and Initiating A Project**

- What upfront work must be done and how much of it?
 - Emergence
 - What constitutes 'enough'
 - Visioning and chartering
 - Sprint zero or discovery phase
 - Starting with a backlog
 - Use workshops and information communication
- **Assess the PRINCE2 Agile Behaviours**

Transparency Collaboration Rich communication Self-organization Exploration



Notes:

1. This would usually trigger an exception
2. This would need attention as it will be causing some damage to the project
3. Although these are green they would still need to be monitored on a regular basis

○ **Assess Level of Uncertainty: Cynefin**

- Decision-making framework to help determine the level of complexity
- Describes the relationship between 'cause and effect'
- Determines how complex an environment is
- Helps to apply PRINCE2 and PRINCE2 Agile correctly

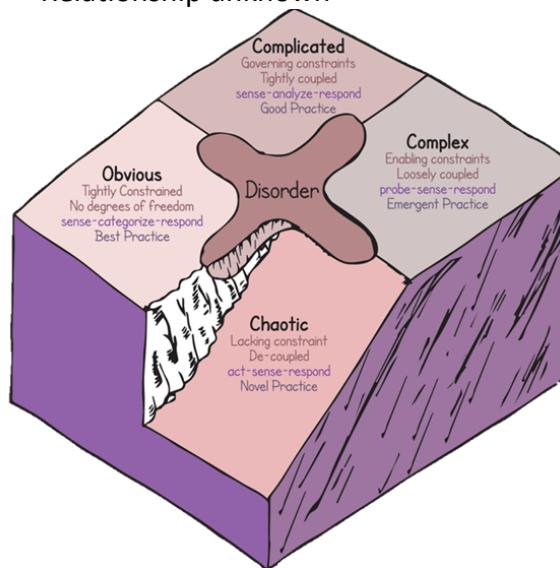
○ **Cynefin Framework**

- Used to analyse two areas
 - Level of complexity of the final product
 - Level of complexity of the project environment
- Helps to better understand the project
 - How should we approach delivery?
 - How should we tailor PRINCE2?
- Projects are usually in the Complicated or Complex domains
- Obvious domain is usually considered BAU
- Chaotic domain is unsuitable for using your existing processes

○ **The Cynefin Framework**

- Obvious
 - Best practice
- Complicated
 - Good practice
- Complex

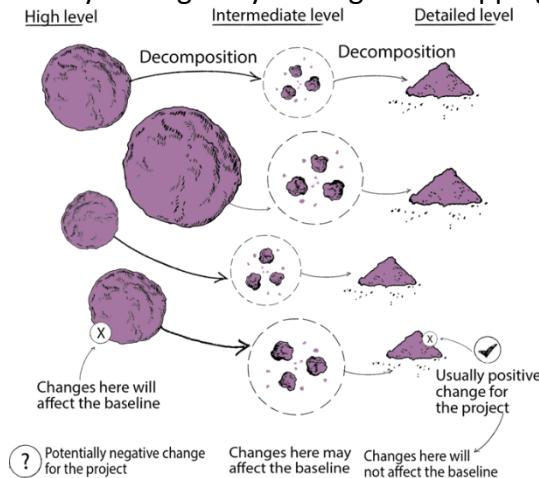
- Emergent practice
- Chaotic
 - Novel practice
- Disorder
 - Relationship unknown



Collaboratively assessed to avoid people's natural tendencies

- **Assess the Agile Environment: Agilometer**
 - 'How much' and not 'yes or no'
 - Use the sliders in isolation
 - Do not add them up or create an average
 - Represents a starting point but it can change
 - After the assessment can any sliders be improved?
- **Directing a Project**
 - **Tailoring Directing A Project**
 - Manage by Exception to empower teams
 - Progress reporting includes the amount delivered and benefits realised
 - Project board attends key demos to gain insight into the details of the project
 - Decision making may be based upon information pulled from information radiators instead of requiring highlight reports
 - **Who Provides the Direction?**
 - Basic Environment
 - Product Owner
 - Basic to Mature Environment
 - Sponsor

- Mature
 - Project Board (Similar to PRINCE2)
- **What to Do...**
 - Manage by exception focusing on
 - Empowerment
 - Amount being delivered
 - Rich information flows
 - Value being delivered
- **How It May Look...**
 - Pulling information rather than having it delivered
 - Collaborative working
 - Trusting
 - A culture absent of blame
- **Requirements and User Stories**
 - Self-directing teams need good requirements
 - Must be well defined and prioritized for agile to work properly
 - Many terms describe what a product does or how well it does it
 - Requirement
 - Product description
 - User story
- **Requirements**
 - Definitions at the right level
 - Decomposed at the right time allowing them to evolve
 - Dynamically managed by trading and swapping detailed requirements

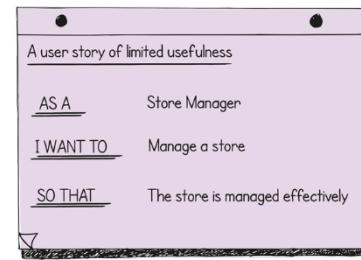
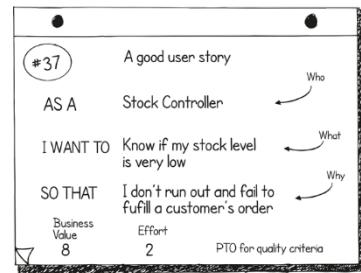


- **User Stories**

As a <role>, I want to <function>, so that <benefit>.

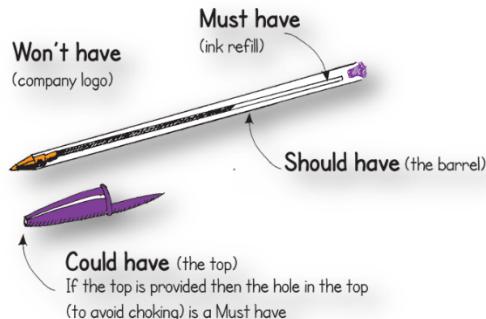
 - Similar to requirements

- Additional information might include...
 - Acceptance criteria
 - Effort involved
 - Value
- A starting point and not fully defined requirements
- Takes skill to make good user stories
- INVEST
 - Independent
 - Negotiable
 - Valuable
 - Estimatable
 - Small (sized appropriately)
 - Testable
- Is it ready?
- **Epic**
 - Type of user story that needs to be broken down
 - These are your boulders...
 - Technical stories can be used for non-functional requirements like performance or speed
- **Prioritization**
 - An essential part of PRINCE2 Agile
 - Two approaches
 - MoSCoW
 - Ordering (1,2,3,...,n)
 - MoSCoW stands for:
 - Must have
 - Should have
 - Could have
 - Won't have for now
- **MoSCoW**



- Must have
- Should have
- Could have
- Won't have for now

Can it be decomposed further?



- **Ordering**
 - Used when there is little dependency between items
 - Items do not naturally group together
- **Controlling a Stage**
 - **Tailoring Controlling A Stage**
 - Stages made up of timeboxes
 - Releases and sprints to create features that enable benefits
 - Team-based collaboration
 - Planning, estimating, flexible work packages
 - Reporting and communication, issues, and risks
 - Stand-ups, information radiators, burn charts, sprint demos
 - Blockers and impediments
 - Agile assessment guides risk management
 - Control focusses on what is being delivered
 - Scope and Quality criteria
 - **What You Might Find...**
 - Higher-level timeboxes
 - Release
 - Iteration or increment
 - More than one team working together
 - Scrum of scrums
 - **What to Do...**
 - Plan around features
 - Create flexible Work Packages
 - Makes management by exception easier
 - Empowers the teams to self-organise
 - Enables rich communication; and is

- Equivalent to a release and multiple sprints or perhaps a single sprint
- Control focuses on what is being delivered
 - Scope
 - Quality criteria
- Monitor agile risks
- **How It Looks...**
 - Collaborative
 - Work assignment is team-based
 - Pulled by the team
 - Visual
 - Information Radiators
 - Stand-up meetings
 - Demos
 - Empirical
 - Inspect and adapt
- **Work Packages**
 - A vital interface
 - Brings PRINCE2 and agile working together
 - Collaboratively defined
 - A clear safe boundary of control
 - Space to empower teams to self-organise and enable rich communication
 - May include one or more releases or sprints
- **Highlight Report**
 - Important yet likely to be informal
 - Contains information on releases and sprints and benefits enabled
 - Could be in the form of an information radiator and/or burn chart
- **Checkpoint Report**
 - Could be replaced by the daily stand-up but must not change the stand-up to “reporting to...”
 - Could be in the form of an information radiator and/or burn chart
- **Managing Product Delivery**
 - **Tailoring Managing Product Delivery**
 - Agile techniques and practice to improve quality and engagement across the delivery team
 - Management frameworks like Scrum
 - Delivery process control frameworks like Kanban
 - **What to Do...**
 - Use work packages appropriately
 - Vital interface and link to be managed

- Define tolerances on what is delivered at the work package level
 - Ensure that product descriptions are defined at the correct level
 - Agree to and provide guidance on areas like quality, releases, and risk
- **How It Looks...**
 - Work packages are collaboratively defined
 - Reporting arrangements have an appropriate level of formality and transparency
 - Wider project context is understood
 - **Prioritization in Delivery**
 - Established during project initiation and used during Managing Product Delivery to...
 - Select the appropriate order of work
 - Manage change
 - Baseline requires formal change management
 - Detailed changes are more dynamic
 - Prioritization needed for both...
 - Quality Criteria (Acceptance Criteria)
 - Functional and non-functional requirements
- **Managing a Stage Boundary**
 - **What You Will Find...**
 - Stages may not exist as described in PRINCE2
 - Viability decisions not usually planned in advance
 - Similarity between a stage and a release
 - Release is a container timebox
 - How are things progressing?
 - How are the team and the processes working?
 - **Tailoring the Process: Managing A Stage Boundary**
 - **Look Back**
 - How did we do?
 - How much was delivered?
 - To what quality?
 - What benefit was delivered?
 - Did the process work well?
 - Release reviews and retrospectives?
 - **Look Forward**
 - Plan the next stage, releases, and sprints
 - Review product and release backlogs
 - Release planning
 - **Look at the big picture**
 - Review the business case

- Review the project plan
 - Review the performance of agile
 - Decide whether to continue?
- **What to Do...**
 - Assess how much is being delivered
 - Assess the quality of what is being delivered
 - What benefits have been realized?
 - Is agile being used appropriately?
 - Do the processes need improving?
 - **How It Looks...**
 - Contains a lot of similarities to Controlling a Stage
 - Visual
 - Empirical
 - Inspect and adapt
 - A point of formality carried out with as little ceremony as possible
- **Closing a Project**
 - **Tailoring a Process: Closing A Project**
 - **Look Back**
 - How did we do?
 - How much was delivered?
 - To what quality?
 - What benefit was delivered?
 - Did the process work well?
 - Final project/release reviews and retrospectives?
 - **Look Forward**
 - How many more benefits can we expect?
 - When will we get them?
 - **Look at the big picture**
 - Check the original baselines against final outputs and outcomes
 - Check products accepted
 - Final operational handovers
 - Documentation finalized
 - **What You Will Find...**
 - Defined processes may only exist in mature agile environments
 - Regular handovers have resulted in activities becoming second nature
 - **What to Do...**
 - Check against the original baseline
 - Evaluate the use of agile on the project
 - Ensure the formality of final acceptance is appropriate

- Finalize documentation that has been created iteratively and incrementally
- **How It Looks...**
 - A workshop is utilized
 - Benefits are already being delivered
 - Majority of information has already been completed

Conclusion

- Conclusion

- Conclusion

- PRINCE2 can be very effective in an agile context
 - Tailoring is about creating an appropriate blend of the two
 - PRINCE2 is already enabled to work with agile
 - Agile covers a wide range of behaviours, concepts, frameworks, and techniques
 - Using agile always asks the question of 'how much can be used according to the situation?'