



Aston University

BIRMINGHAM UK

# Software Project Management

## Unit 4: Agile & Scrum (part 1)

Thais Webber  
Richard Lee



# Goals for this week

1. Describe the core ideas behind agile software development.
2. Begin to explain the structure, participants, artefacts, and events of the Scrum methodology.

# What is agile development?

- Different focus:
  - Change is something to embrace; we determine what is best for the project as we go along
  - Instead of big-design-upfront and fixed deadlines/costs, focus on client-developer collaboration
  - Keep only the parts of the process that deliver value (no process for the sake of process)
- Examples of agile methodologies:
  - **Extreme Programming (XP)** (technical excellence)
  - **Scrum** (management, process)
  - **DSDM (*Dynamic Systems Development Method*)** (controlled agile option)
  - **Crystal** (there is no “one-size-fits-all” Agile method)
  - **Adaptive Software Development** – an evolution of RAD (focus learning)
  - **Pragmatic Programming** (developers encouraged to be flexible but disciplined, mindset)
  - **Kanban** (continuous flow agile, visualise and optimise)

# Projects suited for agile delivery

- The most appropriate projects for agile are ones with aggressive deadlines, high degree of complexity, and high degree of novelty to them

- Novelty



- Urgency



- Complexity



Focus is on delivering the highest business value in the shortest time

# Agile benefits

- Agility allows a better alignment between business objectives and IT
- Agility provides gains in terms of visibility, adaptability, business value, and risk reduction
- Costs reduction thanks to better effectiveness
- Time-to-market reduction
- Alignment with the market and business requirements
- Control over change cost
- Longer life software
- People aware of their responsibilities, more motivated
- Problems arise faster

...but success depends upon people rather than processes

# Agile drawbacks

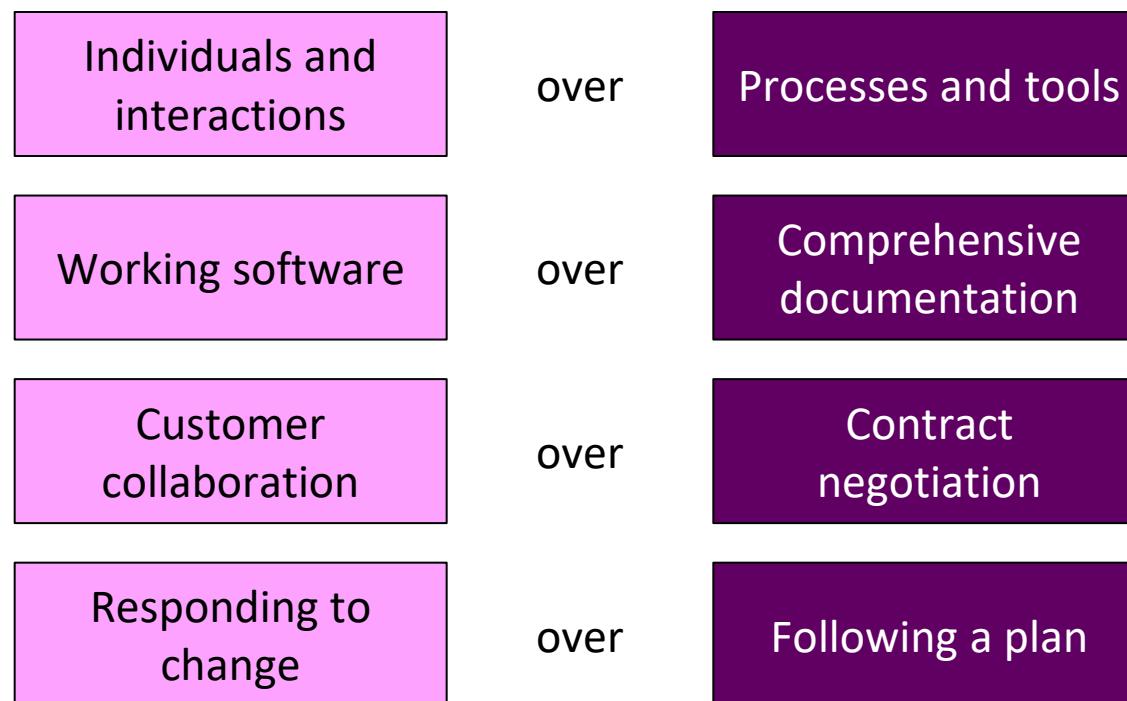
- Active user involvement and close collaboration is essential
- Requirements emerge and evolve
- Frequent delivery
- System structure tends to degrade as new increments are added
- Regular changes to the deliverable usually corrupt the structure unless time & money spent on refactoring

# Quick poll vote – Agile discussion

1 – You prefer to deliver fast and tidy things up later?

2 – You would rather spend more time keeping everything clean from the start?

# Agile Manifesto - values



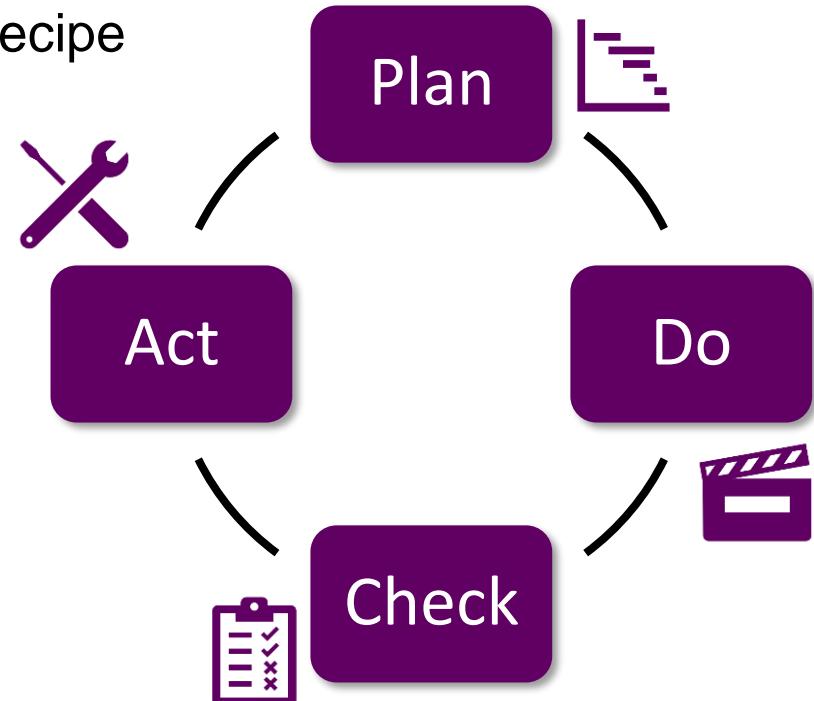
# Other values

- Trust between:
  - Manager and employees
  - Colleagues
  - Customers and partners
- Transparency:
  - To identify and prevent obstacles
  - To be more effective
  - To facilitate collaboration
  - To adapt
  - **Not to watch or oversee**

# Continuous improvement

- At the centre of empiric approaches
- Agility provides a framework, not a recipe
- Adapting practices according to:
  - the product we want to deliver
  - the skills and talents of the team
  - the conditions and the environment
- No agility without quality

(PDCA Cycle)

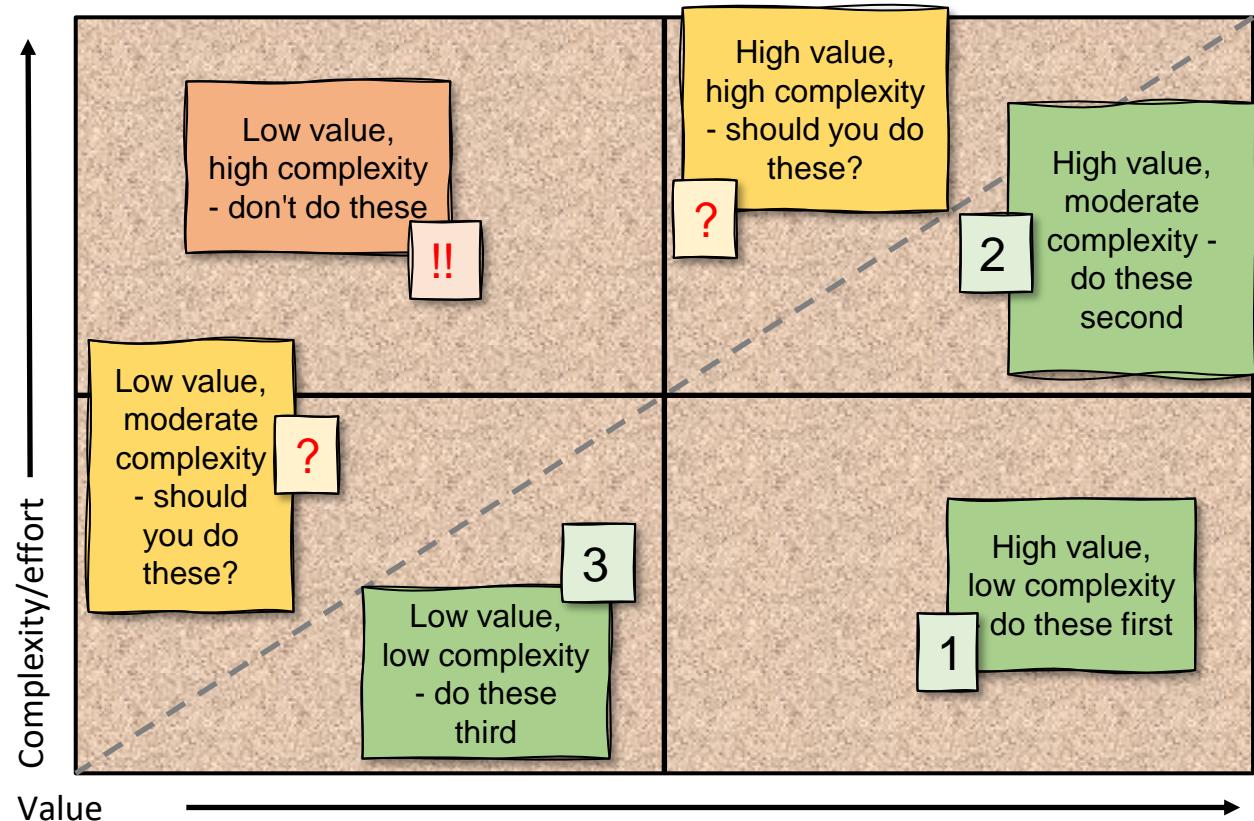


# Sustainable pace

- Working at a steady and sustainable pace:
  - Promotes motivation
  - Increases effectiveness over time
  - Reduces the likelihood of risk
- Traditional project:
  - The closer we are to delivery, the greater the pressure
- Agile project
  - Pace and pressure are steady

# System of prioritisation

- Visibility is central to agile:
  - Information shared with all
  - Problems arise easier
  - Actors feel responsible



- What is Scrum?

- Term originally comes from **rugby**
- Success in a scrum depends upon **working closely** together as a team

- The Scrum framework:

- Lightweight framework that promotes **lightweight, adaptive** solutions to complex problems
- Originated in **1995 (Sutherland and Schwaber)**
- Several books published by both

- Teamwork
- Coordination
- Trust

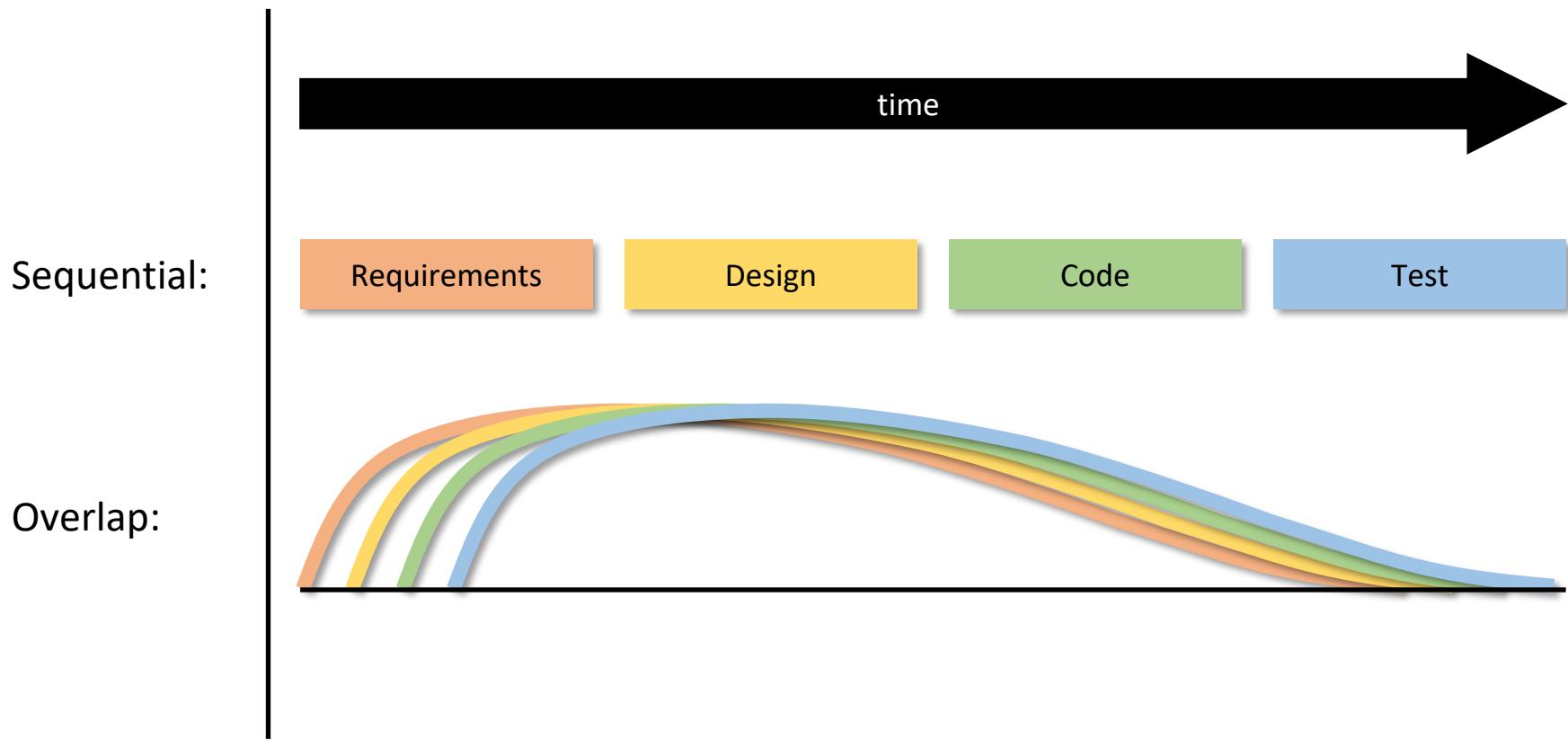
# What is Scrum

- Is an agile, lightweight process
- Emphasis on collaboration
- Can manage and control software and product development
- Uses iterative, incremental practices
- Has a simple implementation
- Increases productivity
- Suited to complex product development
- Reduces time to benefits
- Easy to understand – hard to execute
- Embraces adaptive, empirical systems development

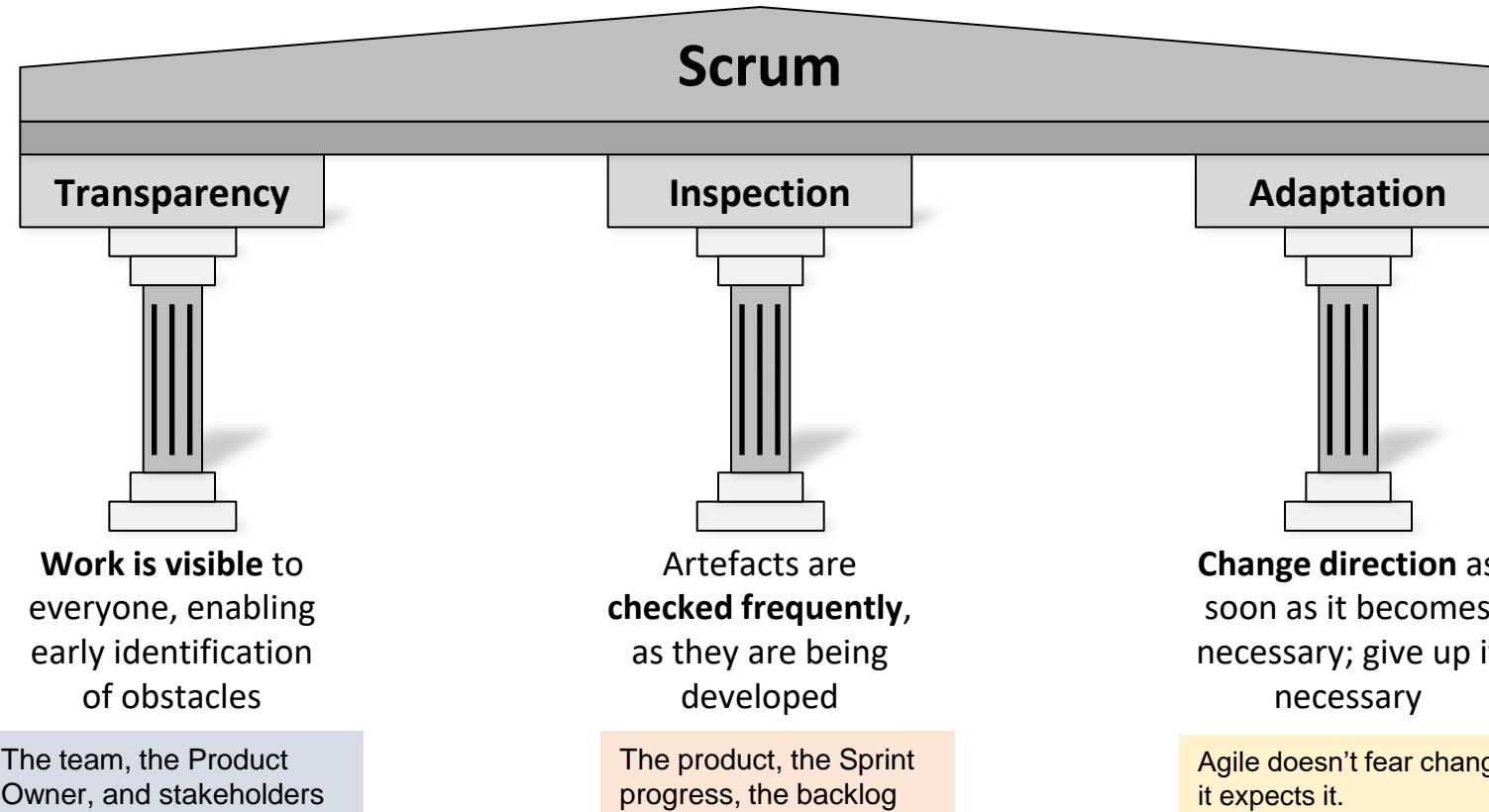
# Values according to Scrum

1. **Commitment** to achieving the goals and supporting each other
2. **Focus** on the work of the Sprint to make the best possible progress toward the sprint goal
3. **Openness** about the work and the challenges you run into
4. **Respect** each other to be capable, independent people
5. **Courage** to do the right thing and work on tough problems

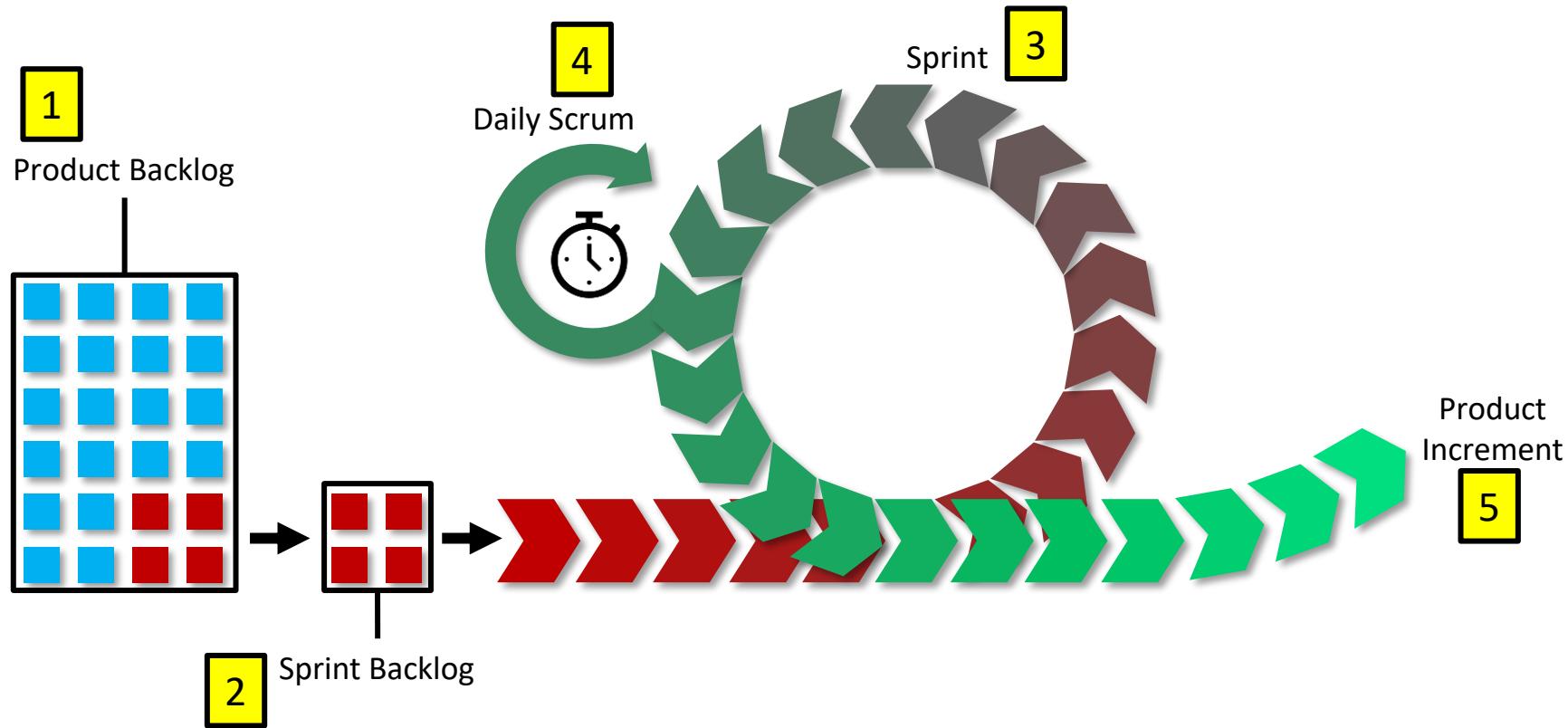
# What is Scrum? Sequential vs. Overlap



# The three pillars of Scrum

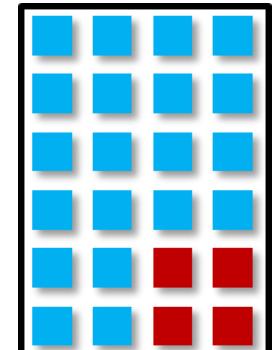


# Product Backlog



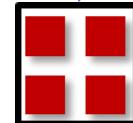
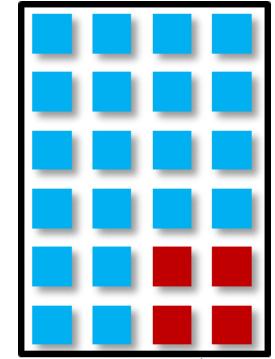
# 1 - Product Backlog

- Defines what will be built
- Contains a broad list of descriptions of all required features, wish-list items, etc. prioritised by business value.
- Product owner is in charge of defining priorities here
- Items here are gradually refined into items that can be completed in a sprint



## 2 - Sprint backlog

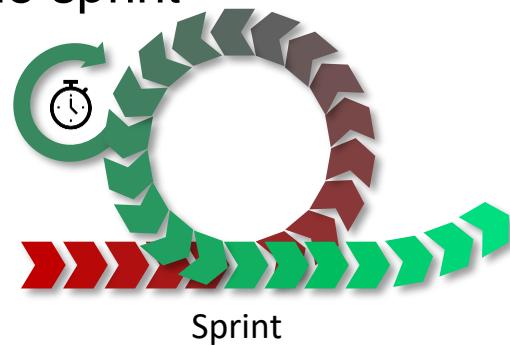
- Task items from the product backlog that will be completed in the next 'sprint'
- Collectively planned by the whole team
- Updated every day to reflect the team's latest decisions



Sprint backlog

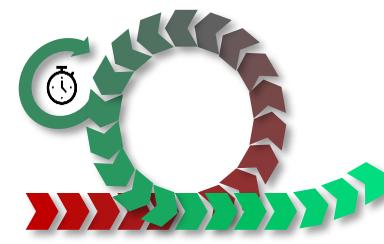
## 3 - Sprint

- Also known as an iteration, these are the units into which large, complex problems are broken down
- Occurs over a timeframe of typically 1-4 weeks, with a new sprint beginning as soon as one ends
- Team must respect the prioritisation agreed for the sprint backlog



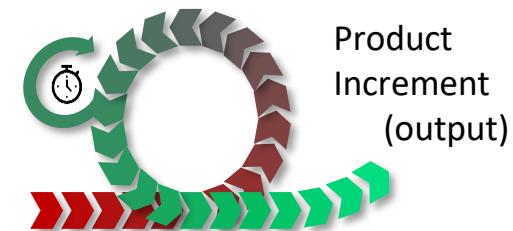
# 4 - Daily Scrum

- Also known as a 'stand-up meeting', this is a short but essential daily meeting involving everyone working on the sprint.
  - Updates on the previous day are shared
  - Plans for the day ahead are shared
  - Anything that has been learned since the last meeting, including problems, is discussed
- Run by an individual titled 'Scrum Master'
- An opportunity to reflect and plan, and ensure work is both synchronised and relevant to the sprint goal



# 5 - Product Increment

- The output of every sprint, which should be a deliverable, functional, usable product
- Each increment builds on all previous increments; the next increment should comprise everything that existed already, plus everything from the latest sprint
- Each increment should be a working version of the product that could, if necessary, be shipped.



fully tested, integrated, and meeting the definition of “done”

# Roles

- **Product owner**

- Possibly a Product Manager / Project Sponsor / Key end-user
- Decides features, release date, prioritisation, budget

maximising value

- **Scrum Master**

- Typically a Project Manager or Team Leader
- Responsible for enacting Scrum values and practices
- Remove impediments / politics, keeps everyone productive

creating space  
for productivity

- **Project Team**

- 5-10 members; Teams are self-organising
- Cross-functional: QA, Programmers, UI Designers, etc.
- Membership should change only between sprints

delivering value  
through collaboration

- **Sprint planning**

- Determining what will take place during a sprint

- **Daily stand-up**

- What did I do yesterday, what will I do today, what obstacles am I facing?

- **Sprint review**

- Everyone who wants to attend can attend; accomplishments are shown

- **Sprint retrospective**

- Examining lessons learned for the next sprint

# FYP and Agile practice – A few tips...

- Apply **Agile thinking** – plan in small, achievable steps
- Be ready to **adapt** when feedback or new ideas emerge
- Keep your **scope realistic** and deliver value early
- **Reflect regularly** – what's working, what needs improvement
- **Document your process** clearly — show evidence of iteration and learning
  - **From Agile:** Product Backlog, Sprint Log or Progress Journal, Reflections
  - **From Waterfall (still useful!):** Project Definition, Requirements document, meetings and decisions log, design diagrams (UML, mock-ups, flowcharts), Timeline (Gantt chart), Trello board – keep versioned documentation



Aston University

BIRMINGHAM UK

# Software Project Management

## Unit 4: Agile & Scrum (Intro)

Thais Webber  
Richard Lee

