Aston University
BIRMINGHAM UK

Software Project Management

Unit 4: Agile, Scrum & Kanban (2)

Thais Webber
Richard Lee

Goals for this week

- Explain the **structure**, **participants**, **events**, and **artefacts** of the **Scrum** methodology

- Understand **Kanban** principles and artefacts

- **Compare** the Scrum and Kanban methodologies, and understand their **advantages** and **disadvantages**

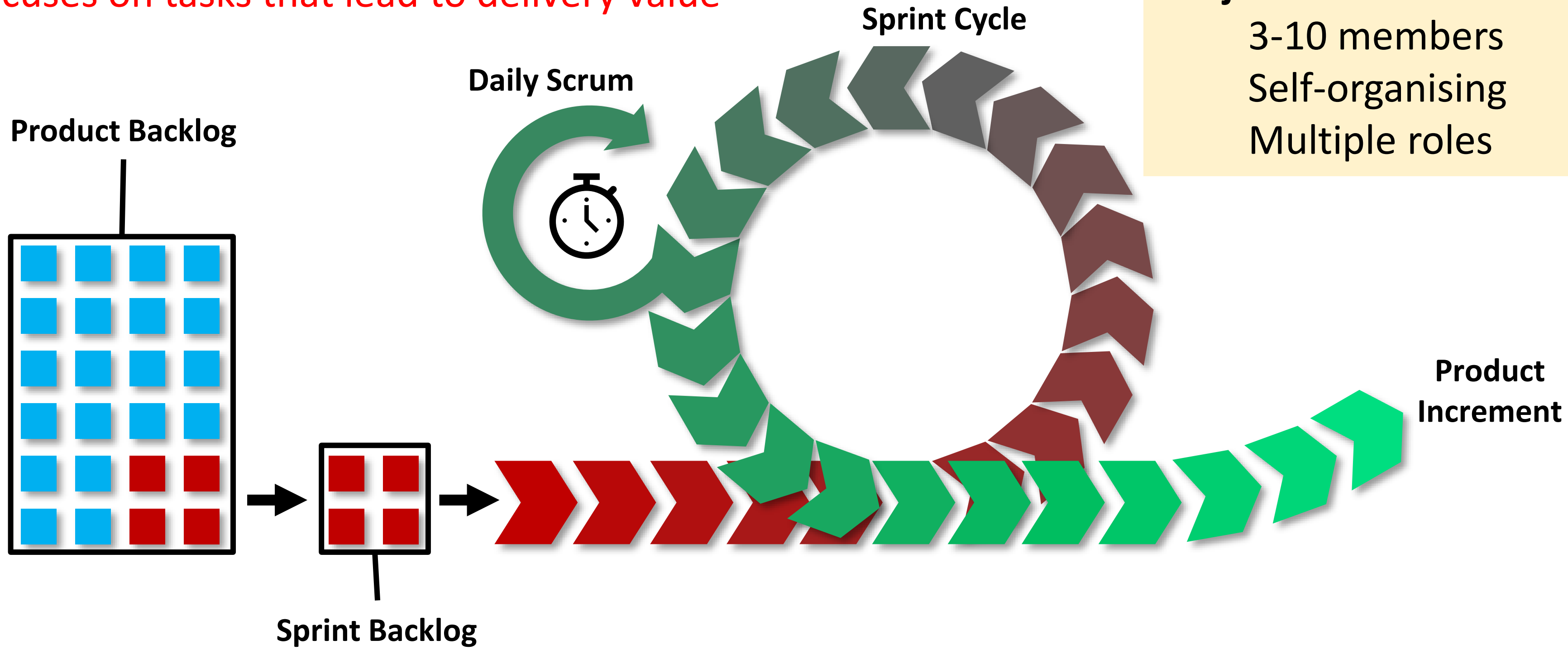
From past lecture… What is Scrum?

- An agile, lightweight framework
- Focuses on **teamwork, collaboration, and adaptability**.
- Uses **iterative and incremental** cycles to deliver value step by step.
- Helps teams **increase productivity** and **deliver benefits faster** through continuous feedback and improvement.

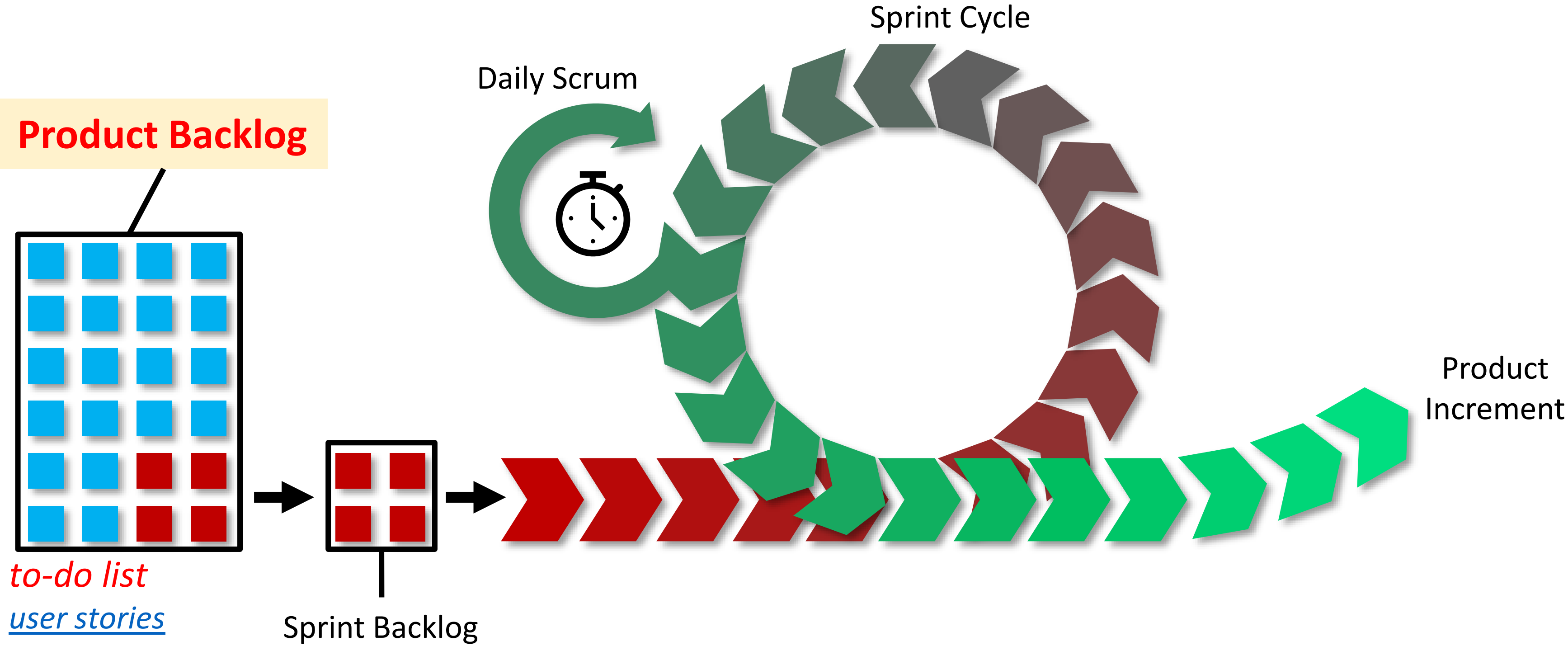
Structure of Scrum

Sprints are fixed time periods in which a team focuses on tasks that lead to delivery value

Project Team
- 3-10 members
- Self-organising
- Multiple roles



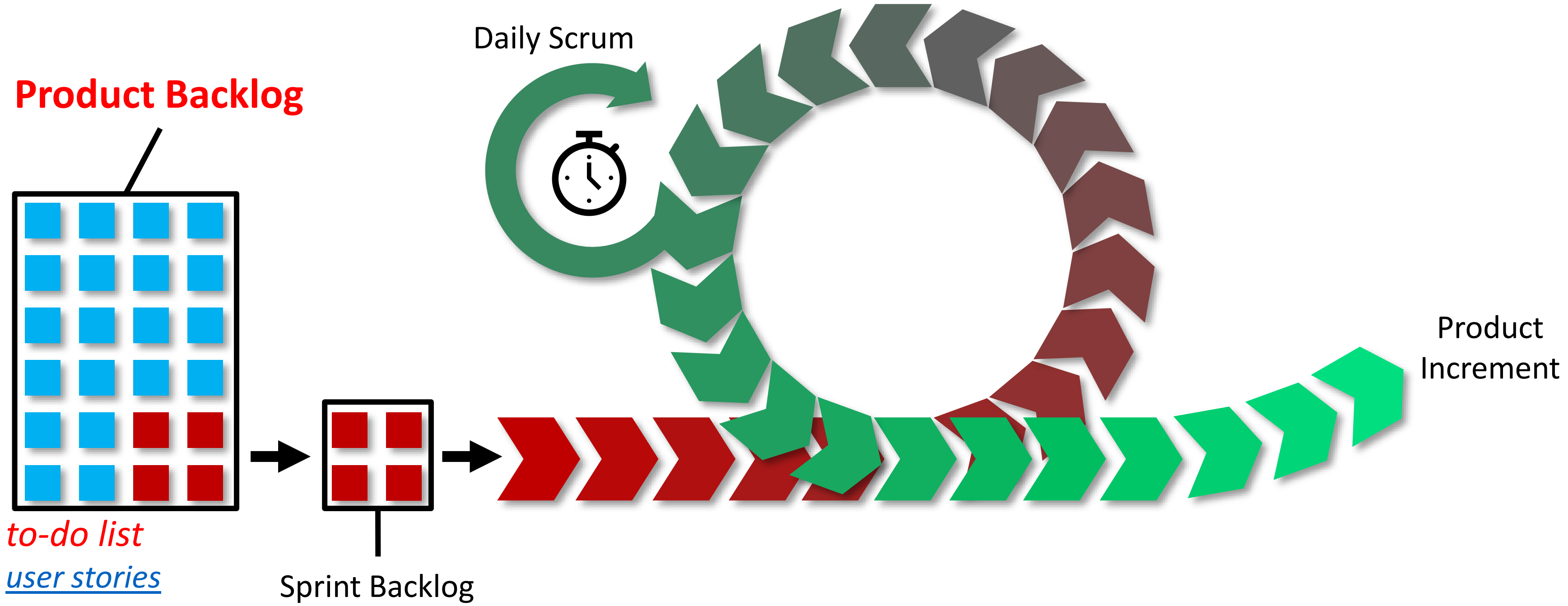
Structure of Scrum



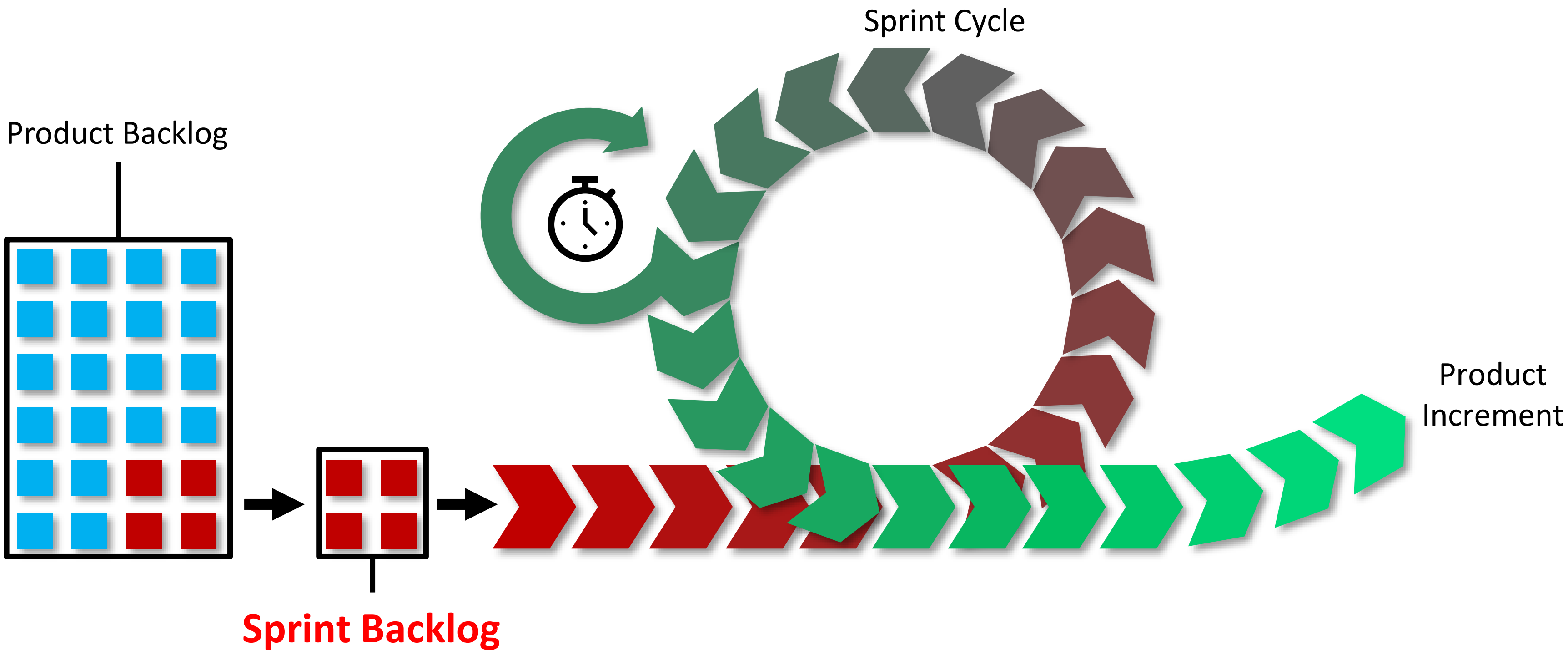
Structure of Scrum

Product Owner

responsible for maintaining the product backlog



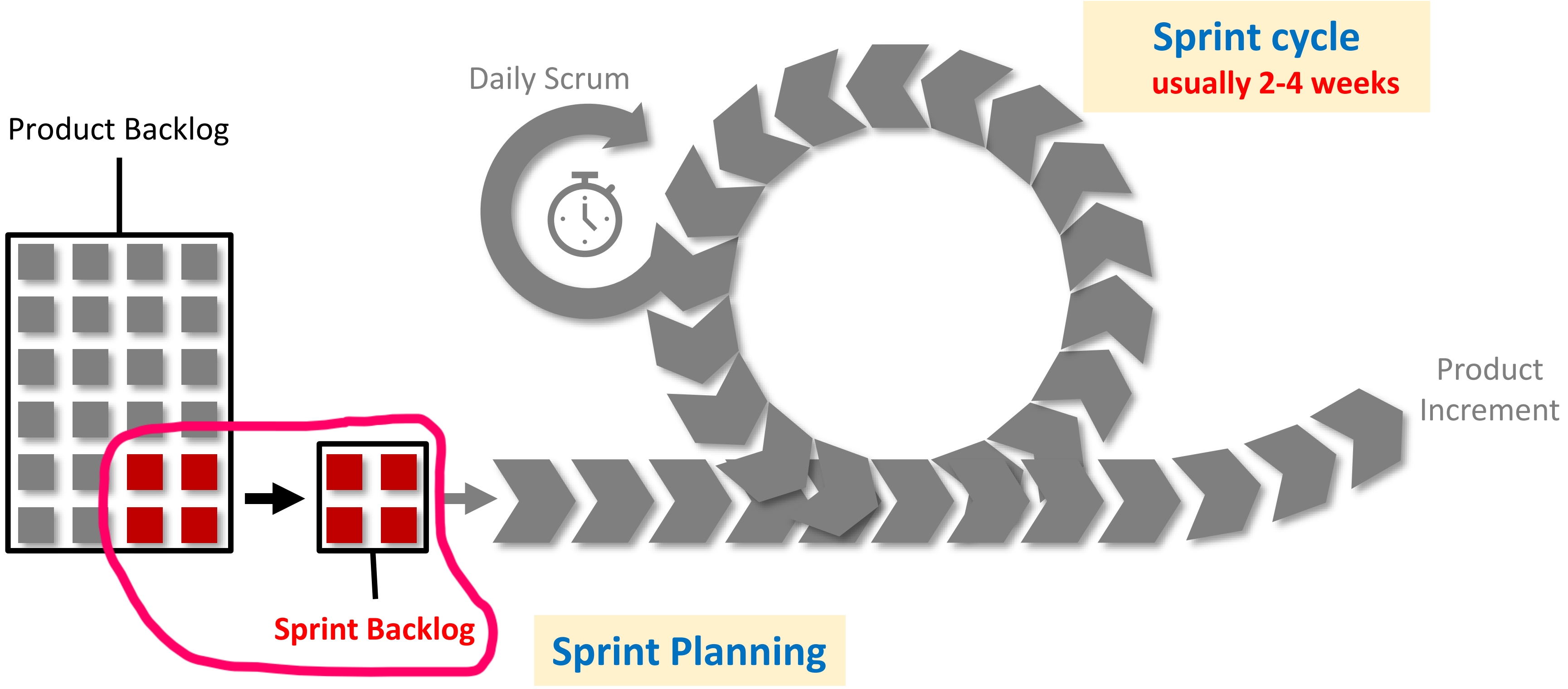
Four key events in Scrum



Sprint Planning

What will take place during a sprint?

Sprint Planning



Sprint Planning

- Provides a short-term vision: from 1 to 4 weeks
- Takes place only at the beginning of an iteration

Objective: set the Sprint backlog (i.e., user stories list) that the team are committed to develop during the sprint according to:

- Effort estimation defined for each user story
- Team capacity and constraints for that sprint

Input: Product backlog

Output: Sprint backlog

Prerequisite: Product backlog is up to date

Sprint Planning

- Purpose:
  - To determine how much work can be accomplished in the upcoming sprint and answer the question 'how will we do it?'

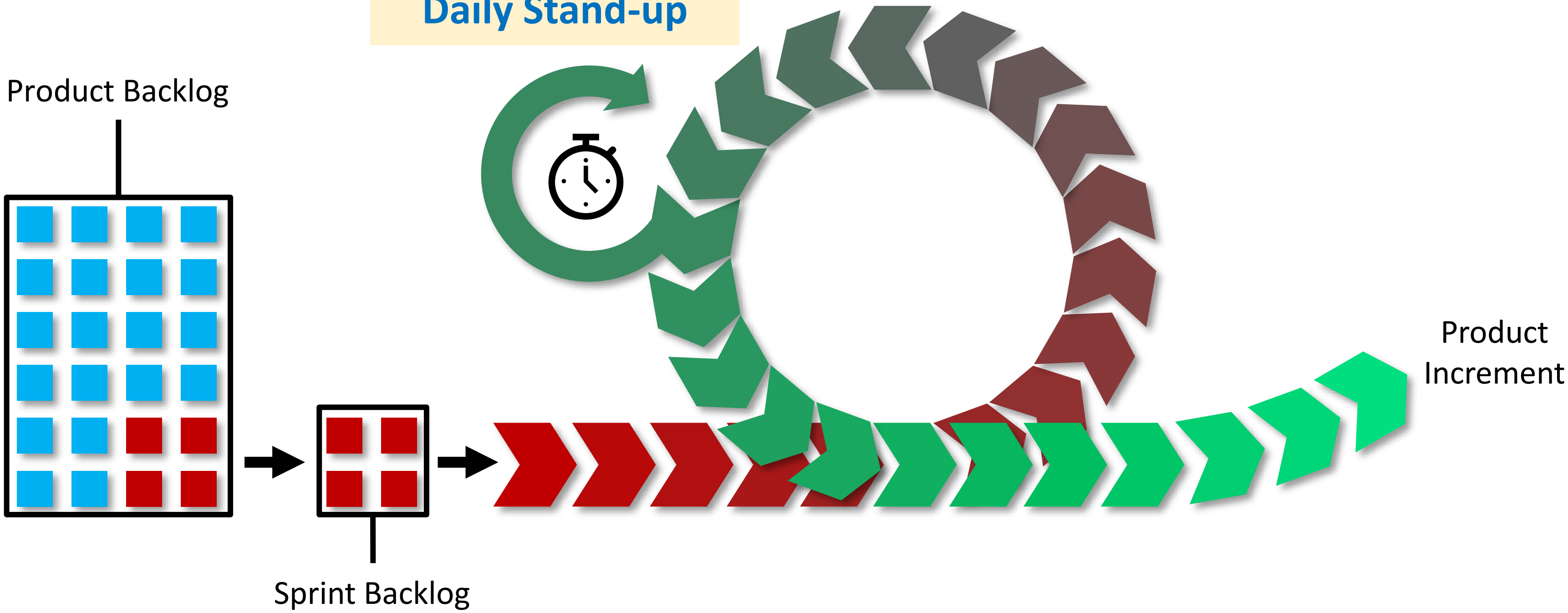
- Outputs:
  - Commitment of several features, stories or bug fixes
  - Sprint backlog describing how to accomplish the commitment

- Rules
  - Product owner must come prepared with clearly defined, prioritised backlog items
  - Product owner has authority over the priorities for the next sprint
  - Team has authority over the amount of work that can be accomplished during the sprint
  - Team must make a commitment before the end of the meeting

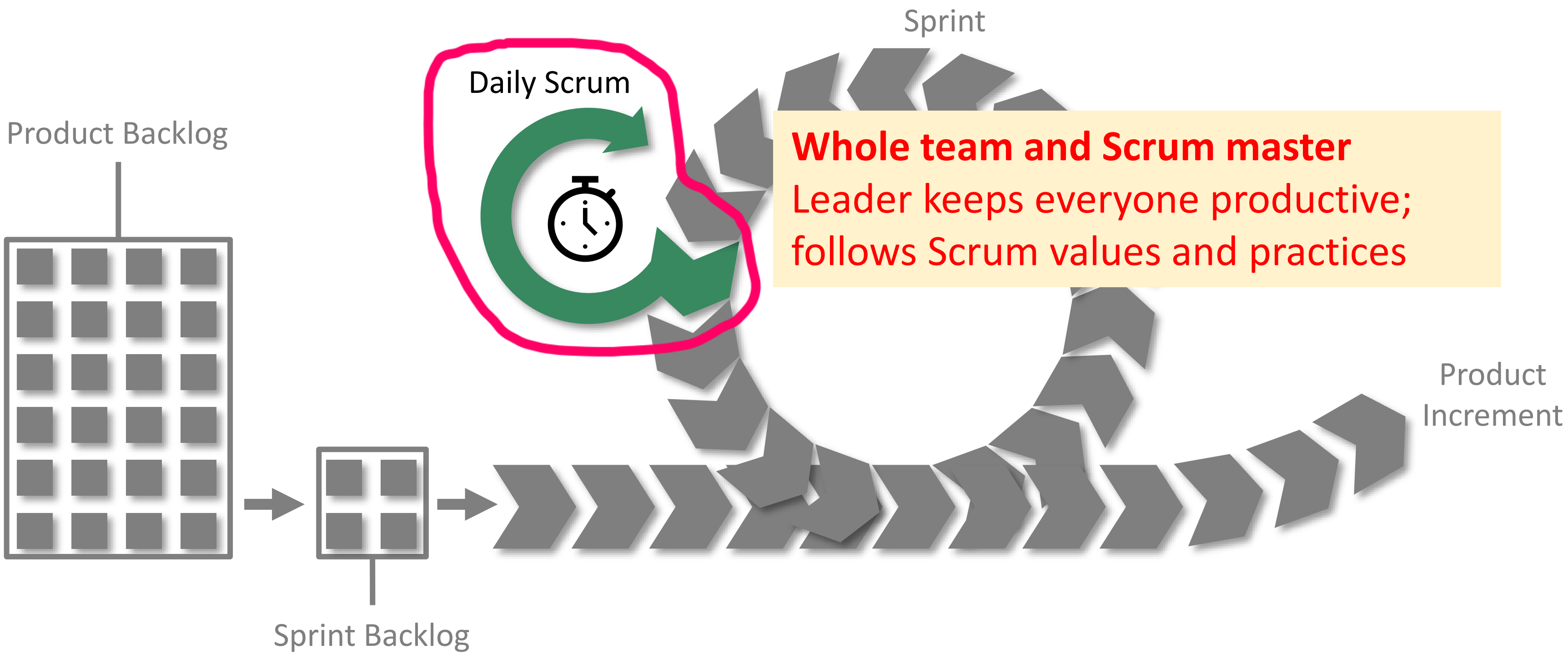
Four key events in Scrum

maximum 10-15 minutes

Daily Scrum or Daily Stand-up



Daily Stand-up



Whole team and Scrum master

Leader keeps everyone productive; follows Scrum values and practices

Daily Stand-Up

- Goal:
  - Enable to team to share progress with each other
  - Make visible the blocks (or impediments) - daily - for whole team to see
- Everyone stands in a circle and reports 3 things:
  - What did I do since the last Daily standup Meeting?
  - What will I try to do by the next Daily standup meeting?
  - Any issues that prevent us to progress?
- 10-15 minutes maximum

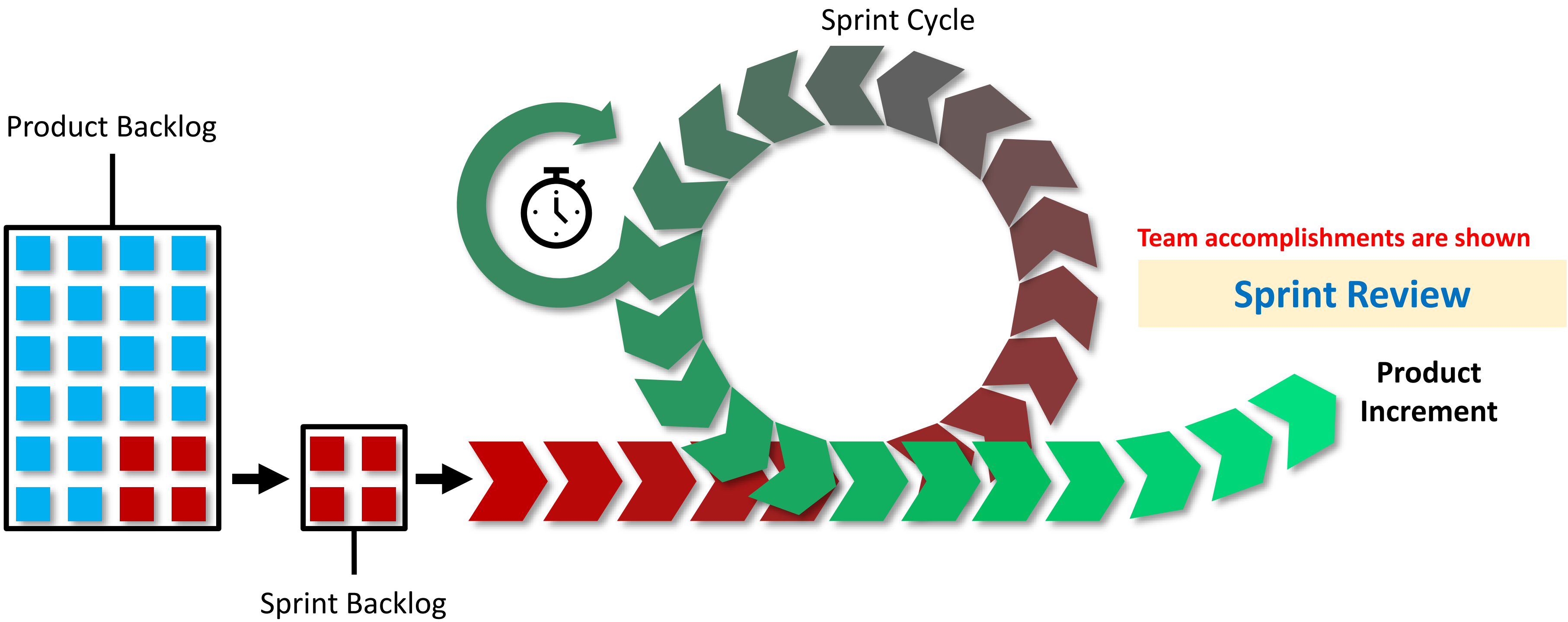
This is the only formal communication each day.

- Informal communication may take place throughout the day on issues or things that need to change (perhaps via Slack or Discord).

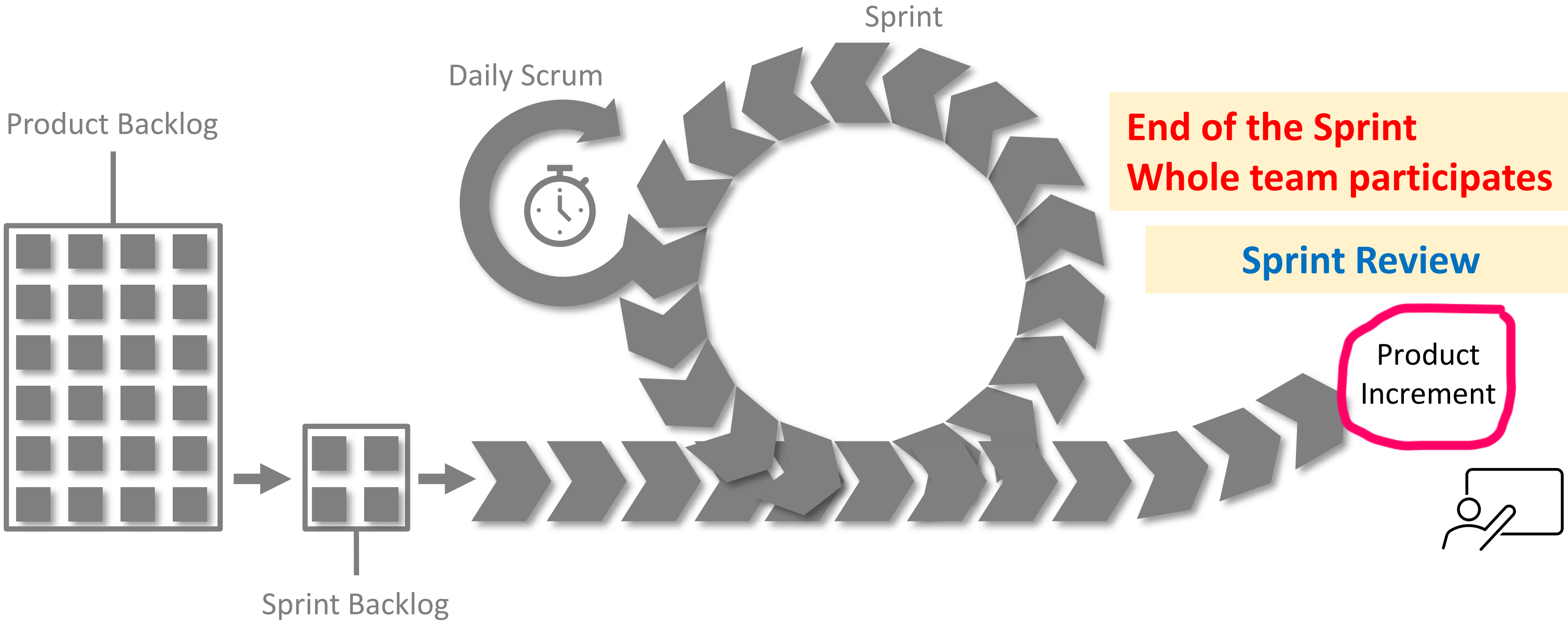
Daily Stand-Up

- No discussion or debate: listening only
  - After the meeting ends, discussion and problem-solving can begin
- Team and Scrum master only
  - Product owner can be invited, as can others, but that's up to the team
- After the meeting, the Scrum Master leads the removal of issues/blocks

Four key events in Scrum



Sprint Review



End of the Sprint
Whole team participates

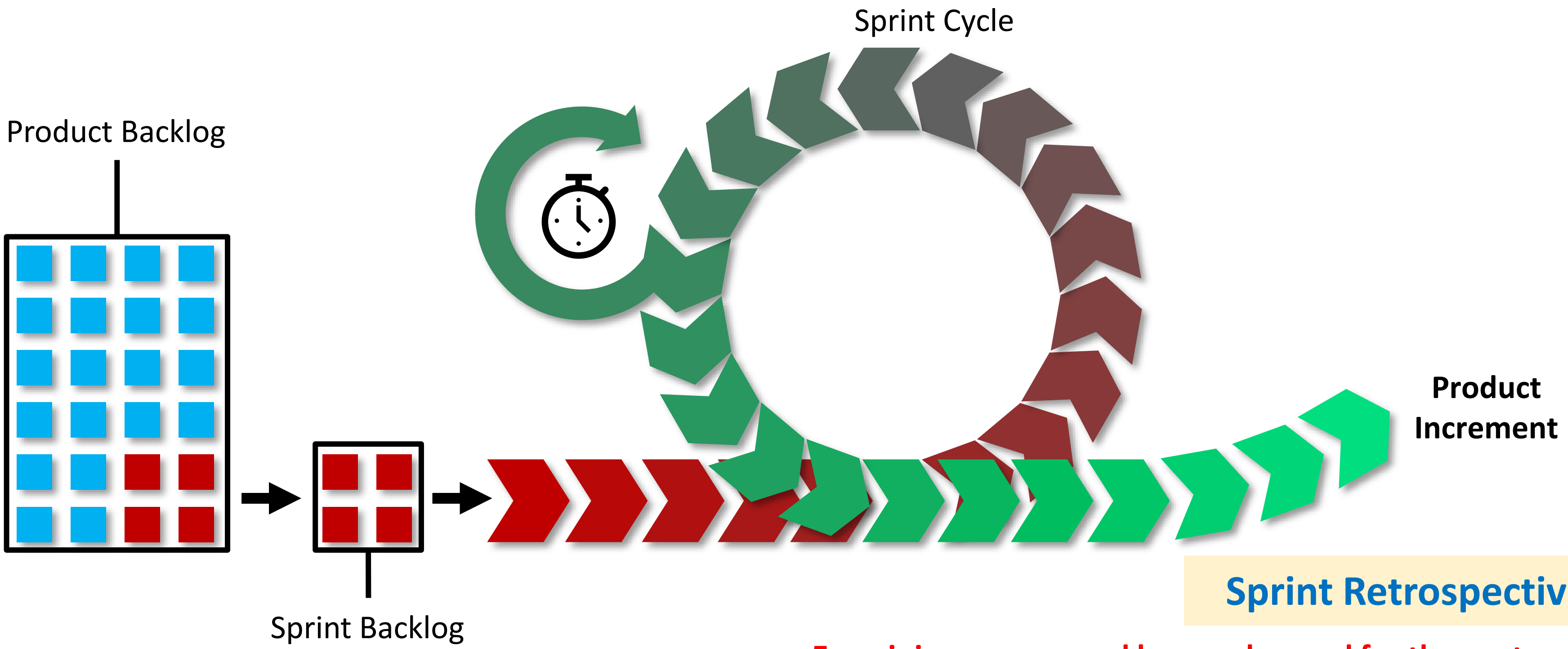
Sprint Review

Sprint Review

- Takes place at the end of the sprint
- Whole team participates
- Only completed user stories (product increment) are presented
- Team presents what it accomplished during the sprint to the business stakeholders and end-users for feedback and ideas

- Typically takes the form of a demo of new features
  - Informal atmosphere
  - Follows 2-hour prep time rule
  - No slides (usually)

Four key events in Scrum



Sprint Retrospective

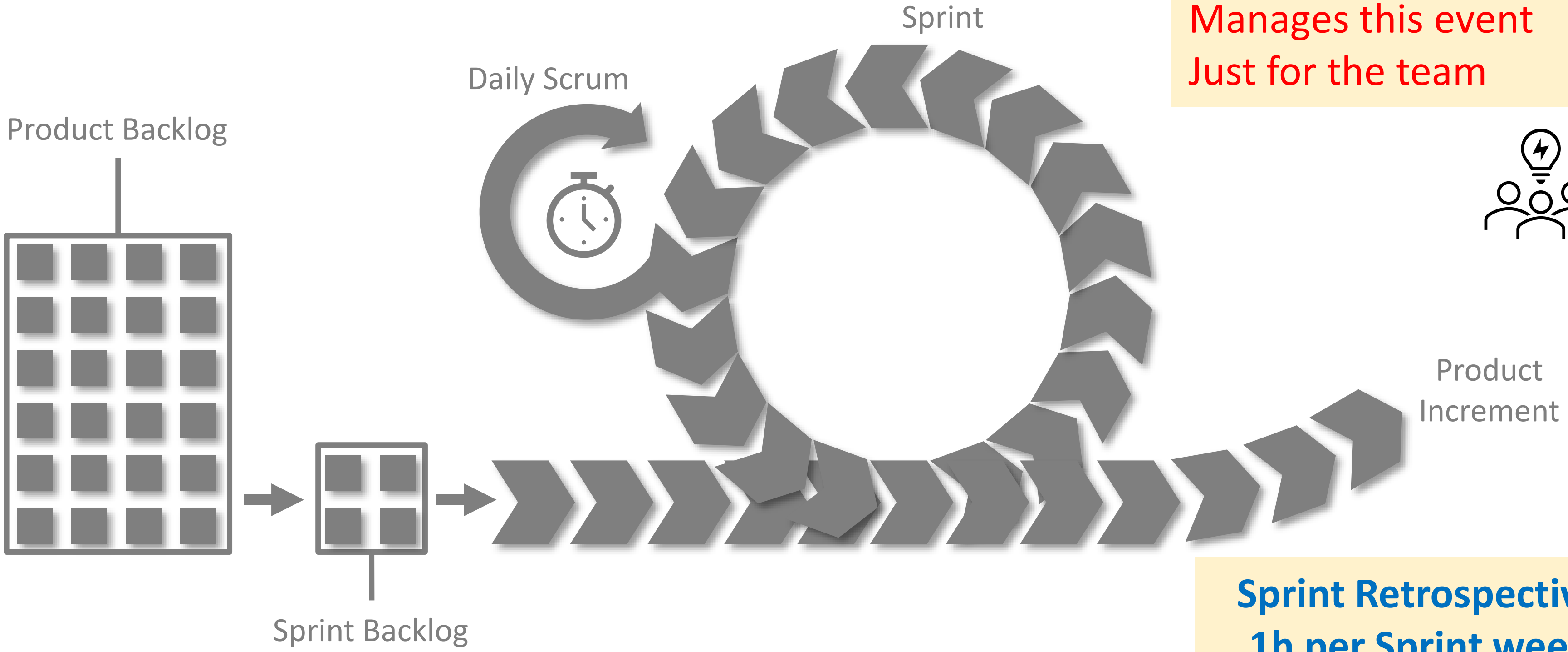
Examining process and lessons learned for the next sprint

Sprint Retrospective

Scrum master

Manages this event

Just for the team



Sprint Retrospective

1h per Sprint week

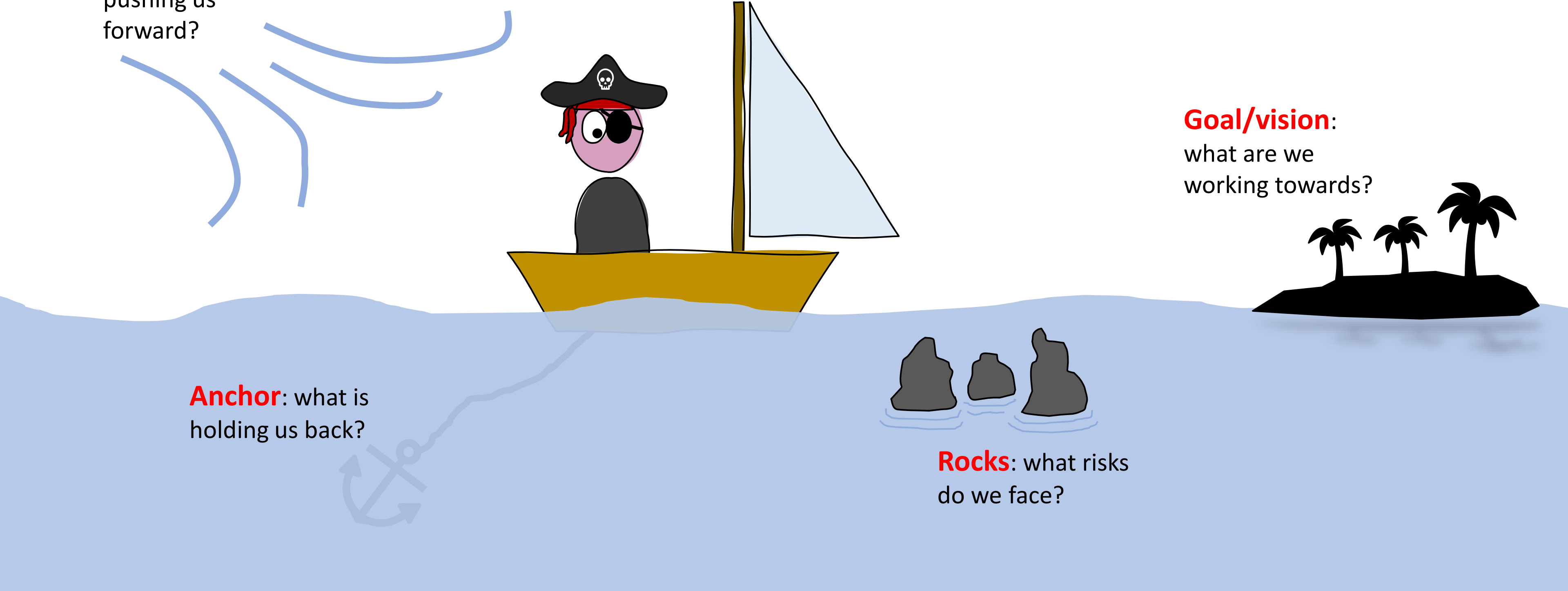
Sprint Retrospective

- Takes place after the Sprint review
- Managed by the Scrum master
- Duration: 1 hour per sprint week
- Period of reflection around Scrum project progression
  - At the organisational level and technical aspects

- Teams can address issues met during the sprint:
  - Issues are identified
  - Issues' importance evaluated
  - Solutions and decisions made

- Format: brainstorming, sailboat, post-it...

Sprint Retrospective (sailboat)



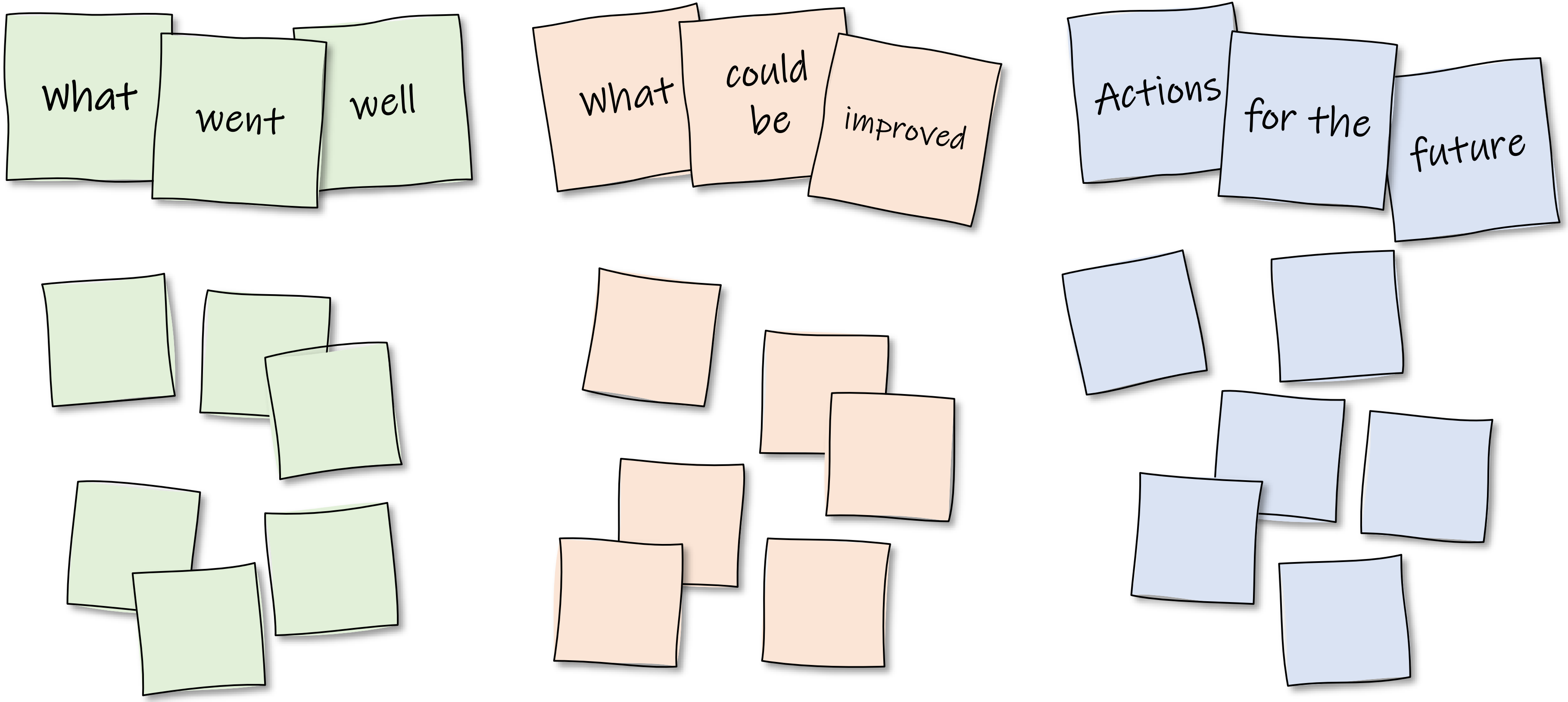
Wind: what is pushing us forward?

Goal/vision: what are we working towards?

Anchor: what is holding us back?

Rocks: what risks do we face?

Sprint Retrospective (post-its)

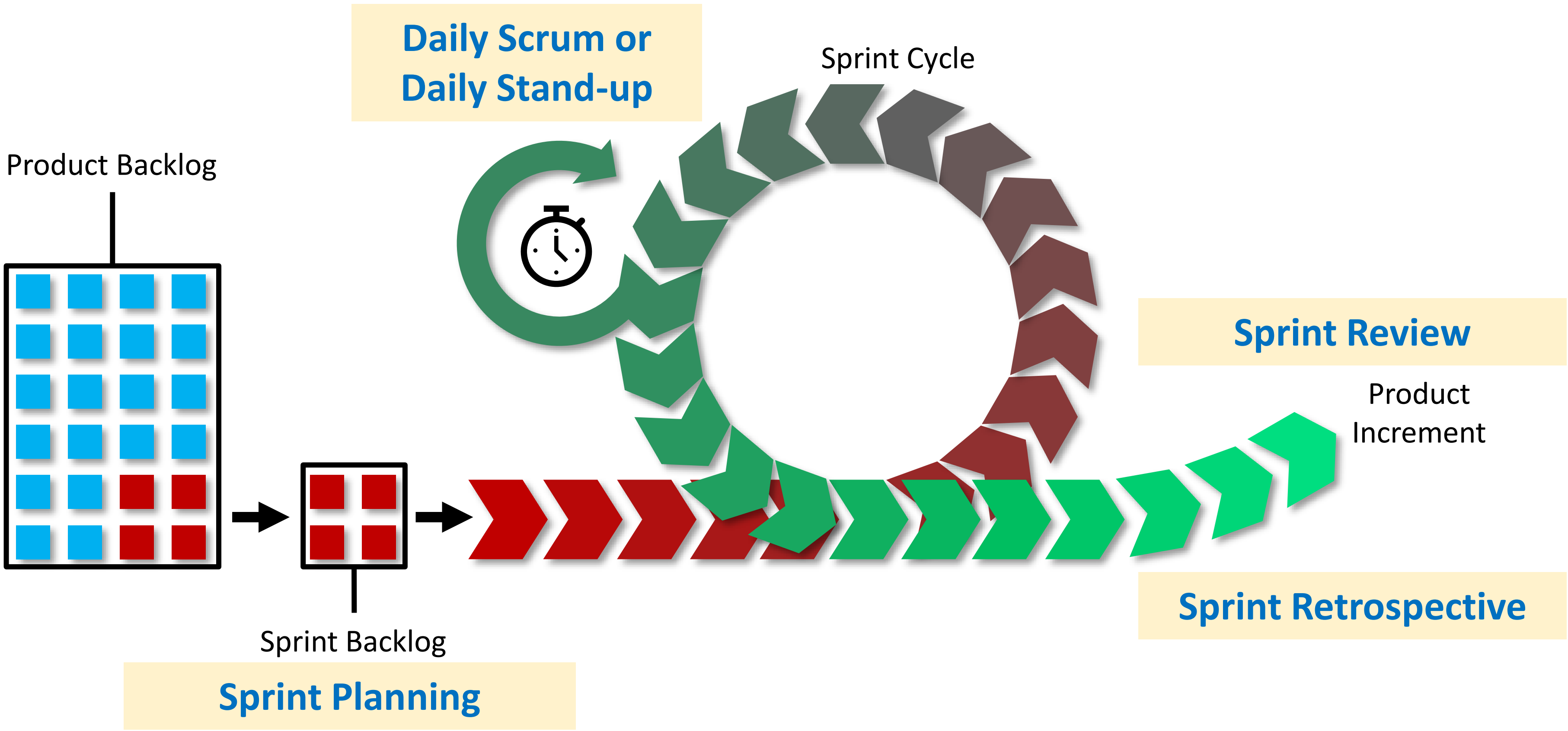


What went well

What could be improved

Actions for the future

Four key events in Scrum



Daily Scrum or Daily Stand-up

Sprint Cycle

Product Backlog

Sprint Review

Product Increment

Sprint Backlog

Sprint Planning

Sprint Retrospective

Events - Summary

- **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

Participants in Scrum

- **Product owner**
  - Possibly a Product Manager / Project Sponsor / Key end-user
  - Decides features, release date, prioritisation, budget
- **Scrum Master**
  - Typically, a Project Manager or Team Leader
  - Responsible for enacting Scrum values and practices
  - Remove impediments / politics, keeps everyone productive
- **Project Team**
  - 3-10 members; Teams are self-organising
  - Cross-functional: QA, Programmers, UI Designers, etc.
  - Membership should change **only** between sprints

Product Owner

- Responsible for the ‘what’
- One product owner per product (avoid mixed directions)
- Responsibilities and activities:
  - Accountable for ROI
  - Engaged with the final product and business users
  - Has a clear vision about the product and have a good understanding of the product
  - Manages the Product backlog
  - Prioritises the functionalities
  - Makes sure that the project team share the same vision and understanding related to the challenges
  - Formalises business needs with different granularity levels according to the importance of functionalities
  - Communicates about progress made

Scrum Master

- Facilitator role
- One scrum master per project
- Responsibilities and activities:
  - Involved in the product implementation
  - Engaged in team progress
  - Manages the project team
  - Removes obstacles that can reduce team effectiveness
  - Makes sure team members collaborate
  - Makes sure the product owner is always feeding the project team

Project Team

- Cross-functional team - includes design, coding, testing, and other resources required for potentially shippable software
- Self-organising and self-managing
- Inspects and adapts through Daily Scrum Meeting and Retrospective
- Assists product owner to prepare the backlog
- Plans the sprint
- Swarms over tasks – minimise Idle work
- Transparent, focused (no more than 2 tasks), works sustainably, stays together

Advantages of Scrum

- Enables projects where the business requirements are hard to quantify to be successfully developed
- Fast moving, cutting-edge developments can be quickly coded and tested
- Due to short sprints and constant feedback, it becomes easier to cope with changes
- Daily meetings make it possible to measure individual productivity. This leads to improvements in the productivity of each team member
- The overhead cost in terms of process and management is minimal, leading to a quicker, cheaper result.

Disadvantages of Scrum

- Scrum is one of the leading causes of scope creep because unless there is a definite end date, the project management stakeholders will be tempted to keep demanding new functionality
- If the team members are not committed, the project will either never complete or fail
- It is good for small, fast-moving projects as it works well with small teams
- This methodology requires experienced team members
- If any of the team members leave during a development, it can have a huge effect on the project development

Scrum – Advantages and Disadvantages

Advantages	Disadvantages
Works well when requirements are unclear or changing	Risk of scope creep if there is no clear end date
Allows fast coding and testing for new ideas	Needs strong team commitment to succeed
Short sprints and constant feedback make change easier to manage	Better suited to small, experienced teams
Daily meetings improve visibility and productivity	Team member turnover can slow or harm progress
Low overhead – lighter processes lead to faster delivery	Requires discipline to avoid confusion or burnout

Introduction to Kanban

- Kanban means ‘signboard’ or ‘visual card’ in Japanese
- Lean method for managing work — not only for software.
- Originally used for just-in-time car manufacturing at Toyota (late 1940s) where parts were manufactured only when demanded (‘pulled’)
  - This reduced work-in-progress stocks and quickly pointed to any bottlenecks hidden by traditional push-oriented schedules
- Adapted for software by David Anderson of Microsoft to handle software maintenance requests

Kanban Steps

1. Capture high-level routines

2. Redecorate your wall (e.g., To Do, In Progress, Done)

3. Define **done**

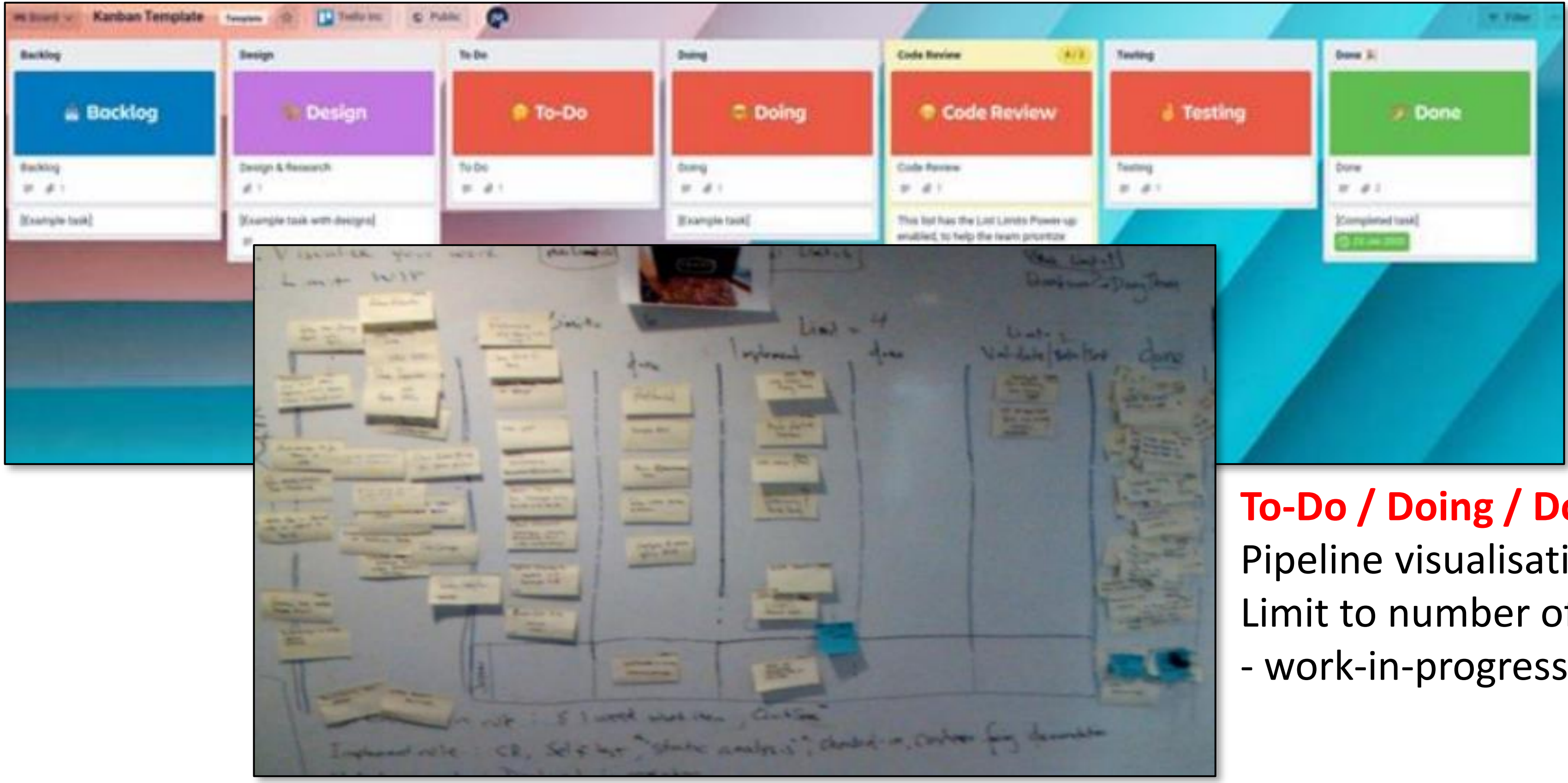
4. Do your daily stand-ups

Kanban: capturing high-level routine

- We need to define the **pipeline** that works goes through
- For instance, when developing new features for your software:

1. Take items from backlog
2. Specify the work to implement it
3. Implement it
4. Validate that it works
5. Deliver to customers or partners

Kanban: redecorate your wall



To-Do / Doing / Done

Pipeline visualisation

Limit to number of cards

- work-in-progress (WIP)

Kanban: define **done**

- Just like Scrum's definition of done
  - 'Done' means the specific requirements for moving a card from one stage to the next

- Examples of 'done' by stage:
  - **Specify** (done): items broken into tasks doable within a day, and quick specs done for each item
  - **Implement** (done): code reviewed and tested, static analysis conducted, code is checked in, customer-facing documentation is completed
  - **Validate** (done): deployment to production, trial by real customers, issues identified are resolved

Kanban: stand-ups

- Project manager asks for any blockers and helps solve these

  (That’s it: there’s no discussion of who’s doing what, as that’s on the board)

- Should take 5-15 minutes
  - Longer discussions can take place separately, involving only those to whom the discussion is relevant

- If a particular step or person keeps getting blocked (or takes longer than expected), further analysis should take place:
  - » Was the work not broken up evenly?
  - » Are there inefficiencies in the process?
  - » Can someone else step in and help?

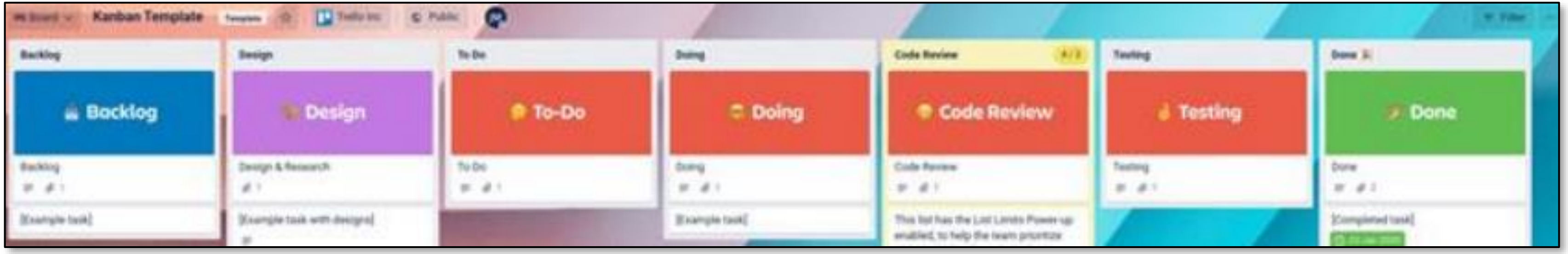
Advantages of Kanban

- **Ease of use**: Kanban is a very simple and easy to understand approach, which makes it practical to apply effectively. No need to be an expert to work with Kanban.
- **Promotes continuous and sustainable improvements**: Kanban approach not only consists of manual cards but also draws visualizations of the process outputs which makes the analysis of work easier.
- **Adaptability**: Kanban encourages maximum adaptability
- **Collaboration**: Kanban focuses on collaboration; this makes the team work together to produce the ideal outcomes/deliverables.
- **Low Overheads**: Supervision of the use of a Kanban board, cards, and analysis of output is easier as compared to most methods/approaches to project management.

Disadvantages of Kanban

- **Lack of iteration**: building software in iterations is a foundation for most development processes, which is not integral to Kanban at a ticket level. You can build iteration on top of Kanban, but it often ends up being its own separate process
- **Lack of timing**: there are no timeframes associated with each phase, which can be disadvantageous
- **Dependency on the board**: if the board is not updated, or is too simplistic to begin with, any advantages that come from Kanban are lost

Kanban – Advantages and Disadvantages



This list has the List Limits Power-up enabled, to help the team prioritize

Advantages	Disadvantages
Ease of use	Lack of iteration
Continuous improvement	Lack of timing
Adaptability	Board dependency
Collaboration	
Low overheads	

Scrum vs Kanban

	Scrum	Kanban
Type	Agile **framework** with defined structure and events	Agile **method** focused on visualising and managing workflow
Key idea	Work in short, **planned sprints**	Maintain **continuous flow** of work
Roles	**Clear roles**; team setup contains Product Owner, Scrum Master, Developers	**No fixed roles**; flexible team setup
Planning	Work is **planned before each sprint**	Work is **pulled as capacity allows**
Focus	Delivering **increments of value** at regular intervals	Keeping **steady progress** and improving flow
Changes	Not allowed during a sprint	Can be made **anytime**

Upcoming sessions overview

- **Next tutorial session scheduled:**
  - **Friday 9:00-10:00** – Reviewing Agile principles and approaches
  - Q&A forums – please your participation is highly encouraged!
- **Coming up next:**
  - **Monday 9:00-11:00**
  - **Risk Management in Software Projects**
    - Learn how to identify, assess, and respond to potential risks before they impact project success.

Aston University

BIRMINGHAM UK

Software Project Management

Unit 4: Agile, Scrum & Kanban (2)

Thais Webber
Richard Lee