universidade de aveiro
deti departamento de eletrónica, telecomunicações e informática

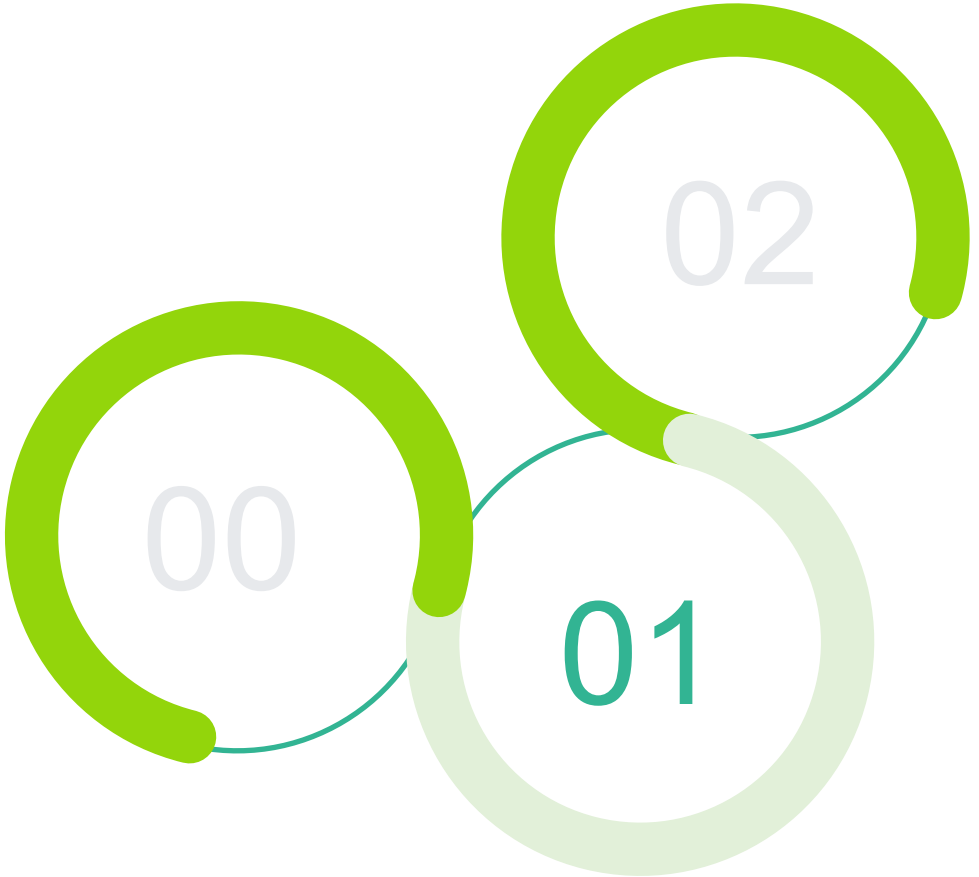
Software engineering principles

Software Engineering

2025/2026

Summary

- Software development process
- Agile methodology
- Scrum
- Project assignment



Project management

- Coordinating tasks, people, and resources to achieve project goals.

- 3 Key pillars
  - **Plan** – Define goals, scope, timeline, resources.
  - **Execute** – Coordinate tasks, guide teams, solve problems.
  - **Deliver** – Meet objectives, measure success, close project.

- Let’s think about building a house…

Project management → Software engineering

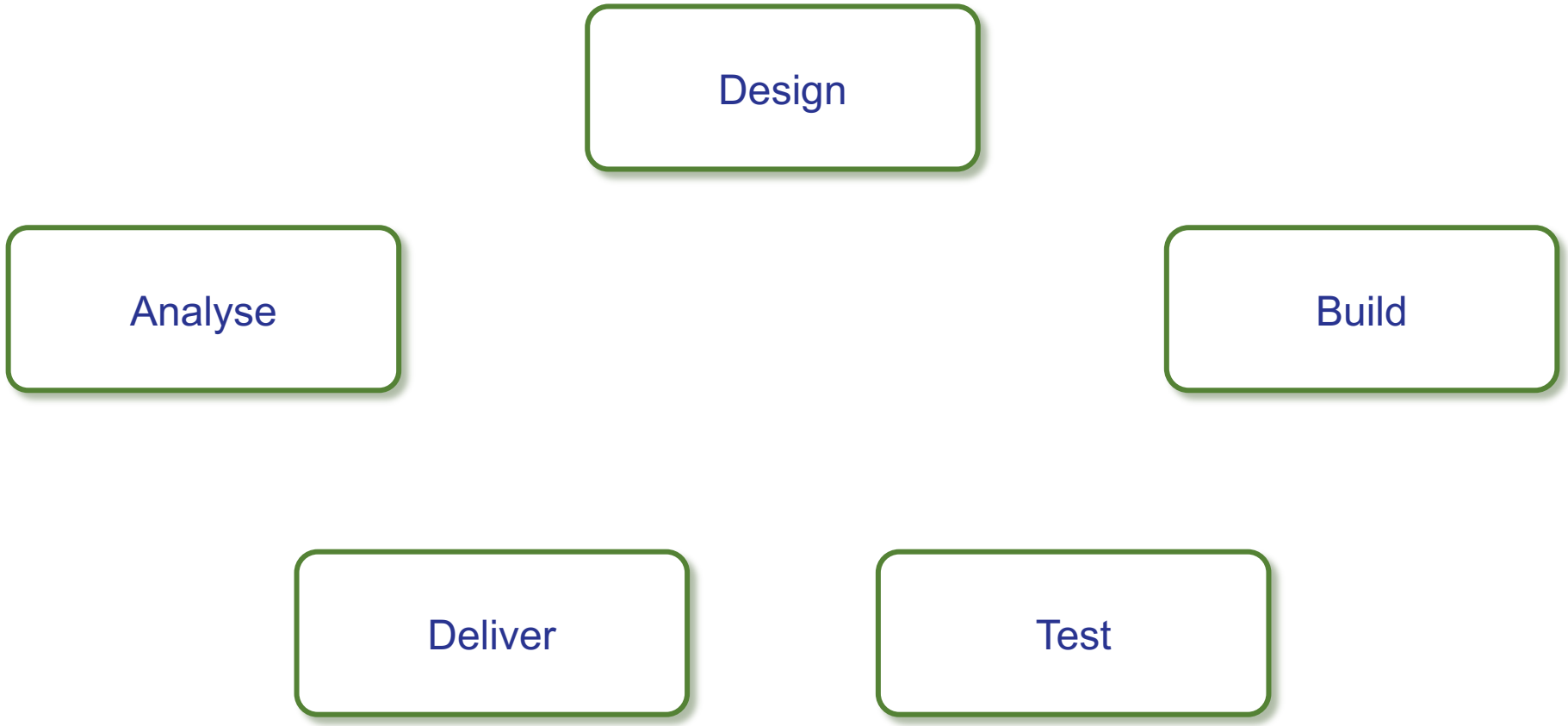
Analyse

Design

Build

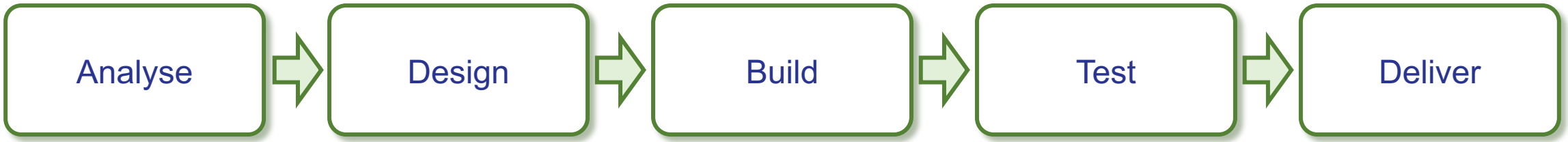
Deliver

Software Development Life Cycle (SDLC)



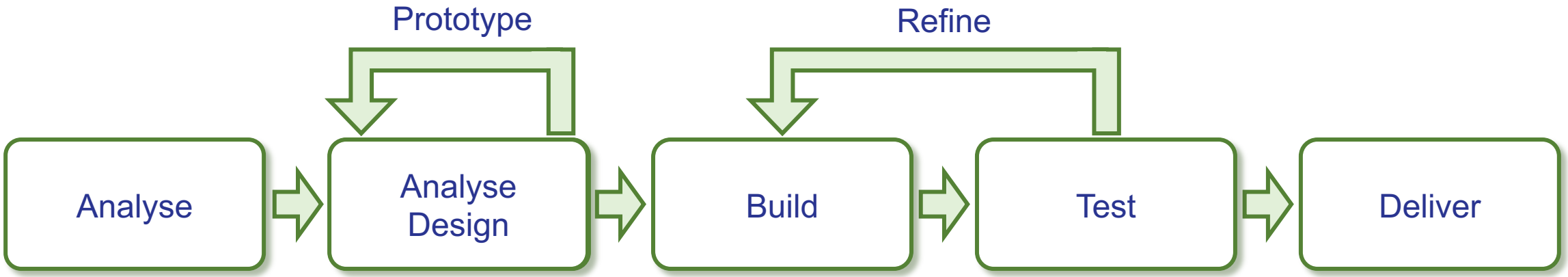
Waterfall

- Plan-driven model
  - Phases are processed and completed one at a time.
- Easy to plan
  - A schedule can be set with deadlines for each stage of development, and a product can proceed through the development process like a car in a car-wash, and theoretically, be delivered on time.
- Works well where requirements are very well understood.



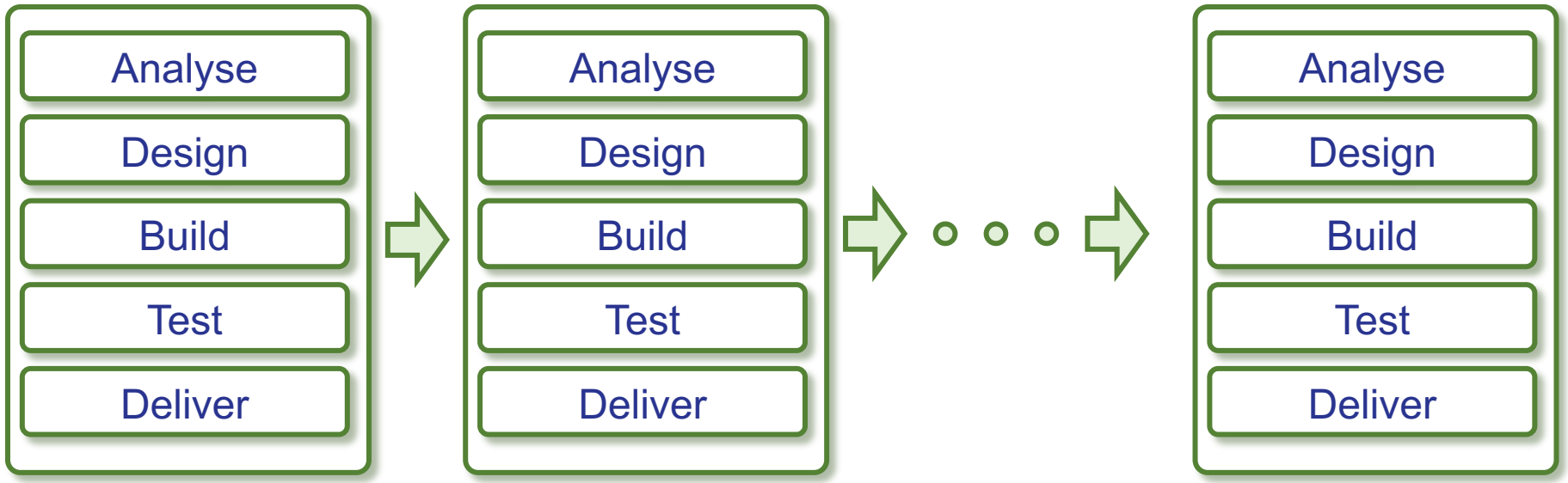
Iterative

- Improve the product through successive **prototypes**
  - incorporating feedback at each cycle, aiming **to learn in each cycle.**
- How it works
  - Build a prototype → Gather stakeholder feedback → Gain team insights.
- When to use:
  - High **complexity;** Frequent **changes;** Different **stakeholder views** on the final product.



Incremental

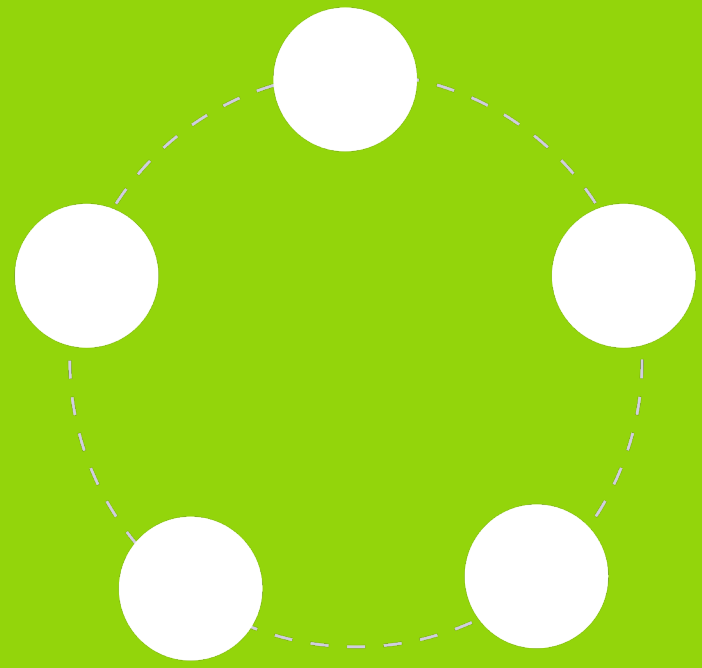
- Work is optimized to deliver **value to sponsors/customers** earlier and more often, instead of waiting for a single final product.
- Customers get **value sooner**, even if the final vision evolves.



Characteristics of project life cycle

Approach	Requirements	Activities	Delivery	Goal
Predictive	Fixed	Performed once for the entire project	Single delivery	Manage cost
Iterative	Dynamic	Repeated until correct	Single delivery	Correctness of solution
Incremental	Dynamic	Performed once for a given increment	Frequent smaller deliveries	Speed
Agile	Dynamic	Repeated until correct	Frequent smaller deliveries	Customer value via frequent deliveries and feedback

Agile methodology



Origin of agile manifesto

- On February 11-13, 2001, 17 software leaders from diverse methodologies met to find common ground
  - Frustrated with heavy, documentation-driven processes, they sought an alternative.
- The concept “Agile” was born in the first day
  - a shared set of values focused on people, collaboration, adaptability, and trust.
- The next day the **Agile Manifesto** was written…
  - …shortly after 9 participants stepped out for a smoke break.

Agile manifesto

We are uncovering better ways of developing software by doing it and 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***

The 12 agile principles (1/4)

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 timescale.

The 12 agile principles (2/4)

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 12 agile principles (3/4)

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.

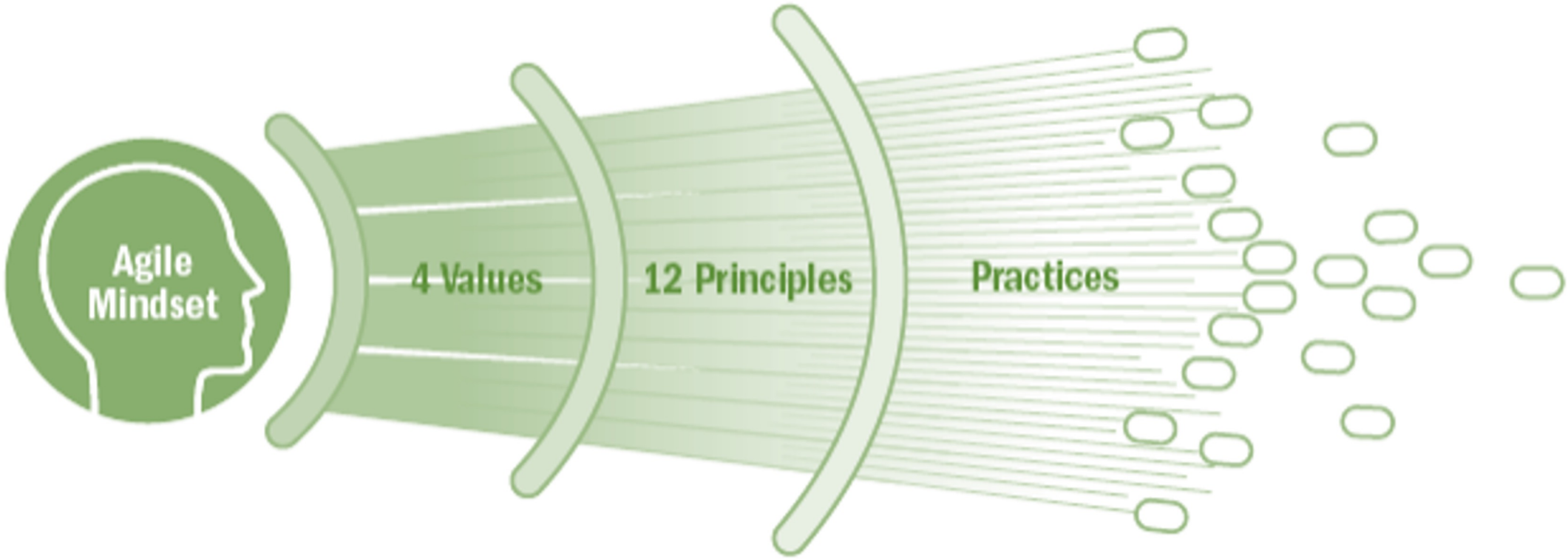
The 12 agile principles (4/4)

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.

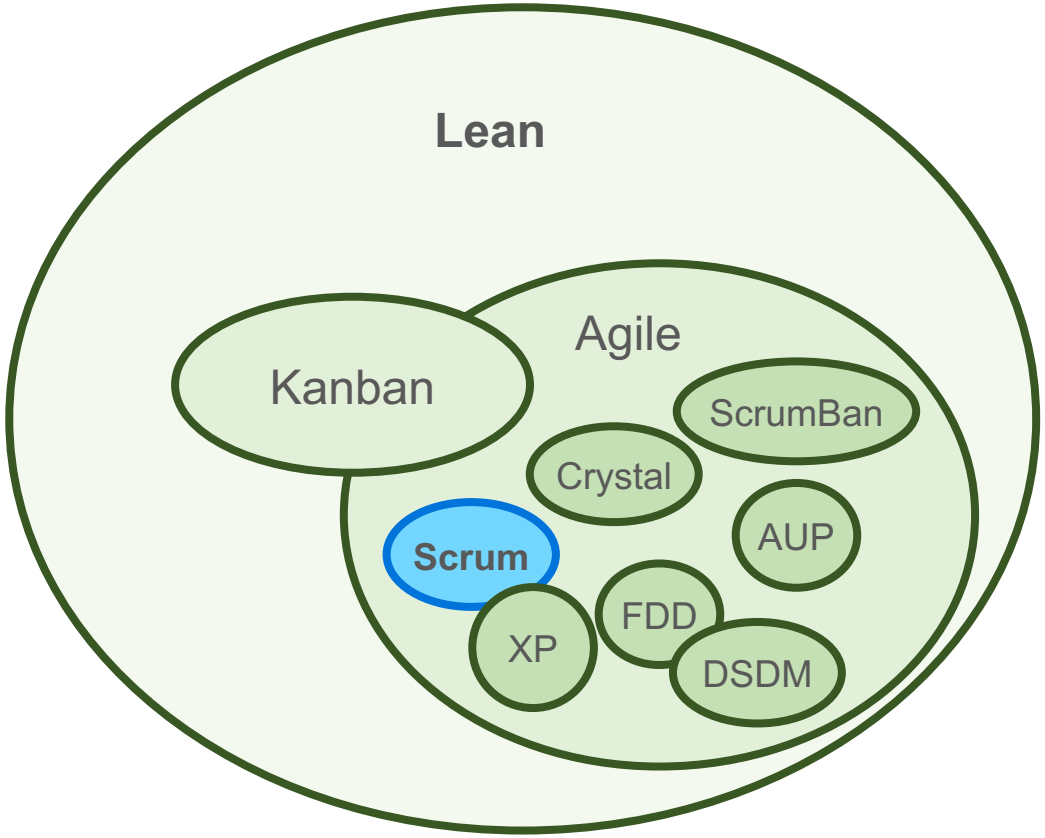
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile mindset



Agile management methodologies

- Agile methods are umbrella terms that cover a variety of frameworks and methods
- It shares the lean thinking
  - “focus on value”
  - “small batch sizes”
  - “elimination of waste”
- We will focus on Scrum
  - 63% of Agile users use Scrum
    - based on 17th annual State of Agile Report

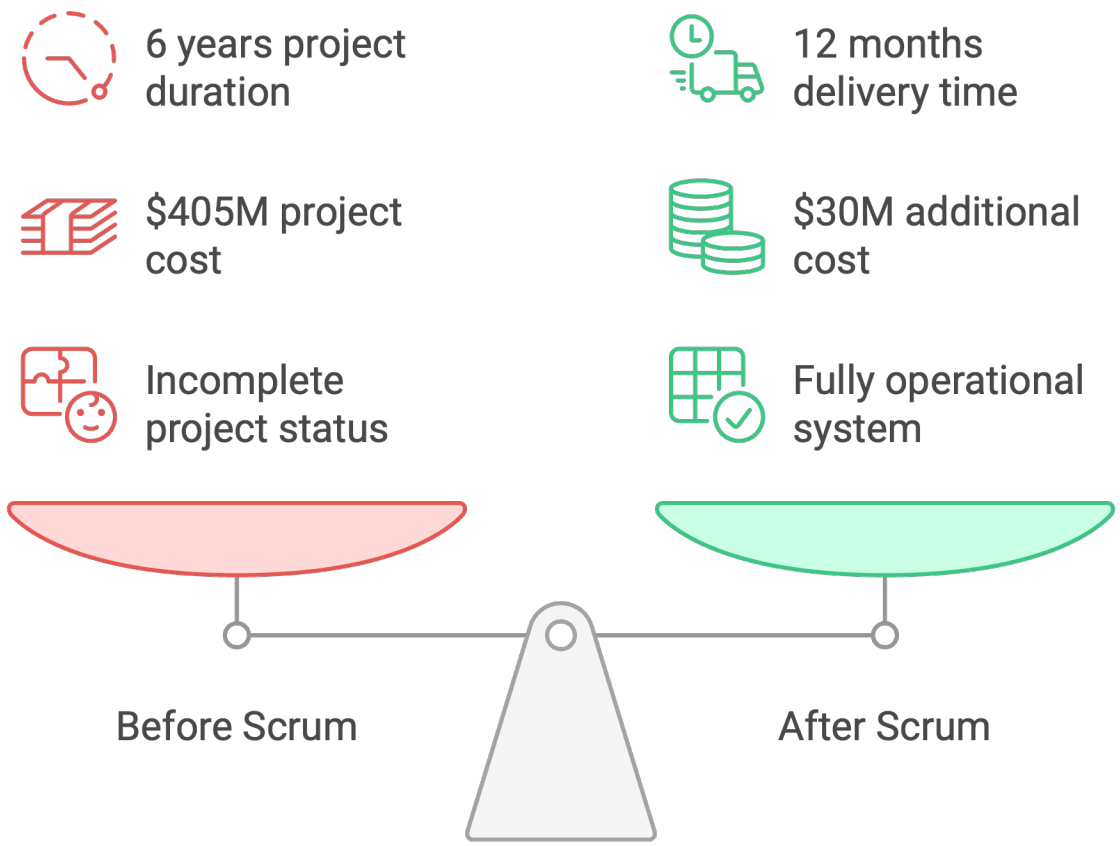


https://stateofagile.com/

Scrum

FBI Sentinel project

- A new case management system to replace FBI outdated infrastructure
- After **six years** and **$405 million spent**
  - the project was nowhere near completion
- Jeff Sutherland, co-funder of Scrum, uses Scrum to solve the problem
  - And to tell the story how Scrum can be helpful



Scrum definition

- Scrum is a lightweight framework that helps people, teams and organizations **generate value** through **adaptive solutions** for **complex problems**.

- Key points:
  - Scrum is **simple** but purposefully incomplete.
  - Builds on the **collective intelligence** of those using it.
  - Provides **guiding rules**, not step-by-step instructions.
  - Can wrap around existing practices or make them unnecessary.
  - Makes visible the **effectiveness** of current methods to enable improvement.

Scrum theory

Transparency

Ensures visibility and understanding of processes and artifacts.

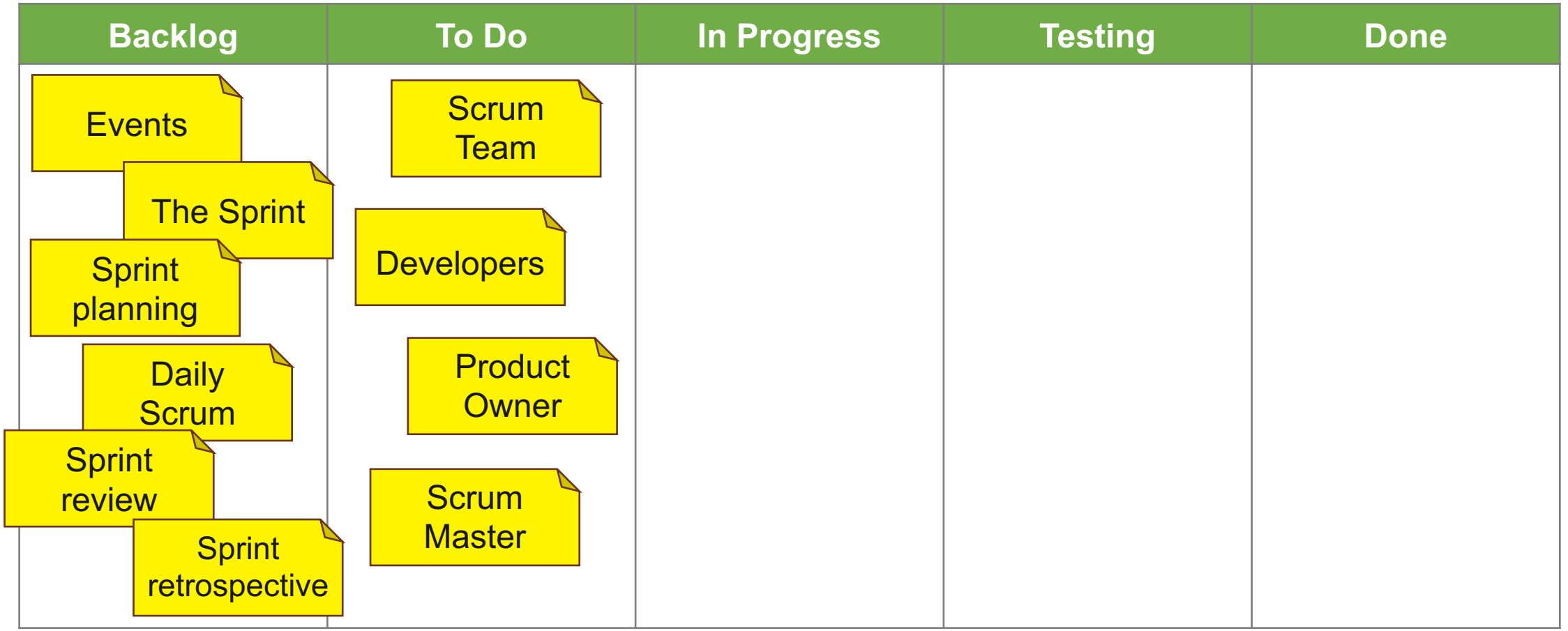
Inspection

Involves frequent and diligent review to detect issues.

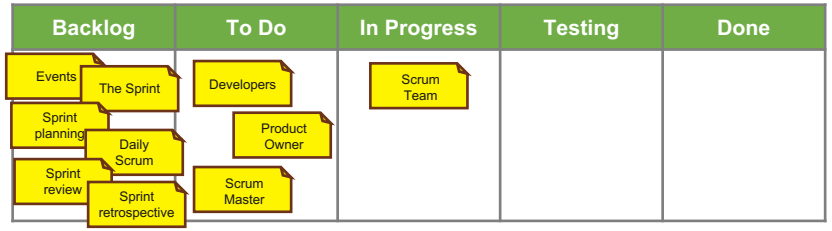
Adaptation

Requires timely adjustments based on new insights.

Learn Scrum with Scrum – Sprint #1



Scrum team



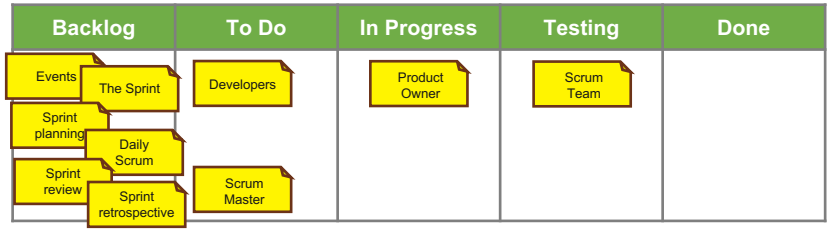
- **No hierarchy**
  - Unified, cross-functional, self-managing
- **Size**
  - Up to 10 people for agility and productivity
- **Shared goal**
  - One Product Goal per Sprint
- **Scope**
  - Full ownership — development, collaboration, maintenance, R&D
- **Accountability**
  - Entire team delivers a valuable Increment every Sprint

Product owner

Scrum master

Developer

Scrum team – product owner



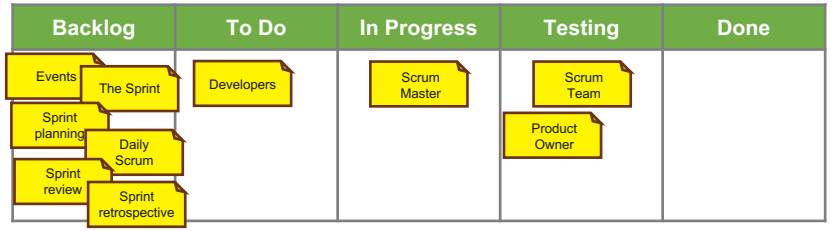
- The Product Owner is accountable for **maximizing the value of the product** resulting from the work of the Scrum Team.
  - How this is done may vary widely across organizations, Scrum Teams, and individuals.

- Product Backlog management, which includes:
  - Developing and explicitly communicating the Product Goal;
  - Creating and clearly communicating Product Backlog items;
  - Ordering Product Backlog items; and,
  - Ensuring that the Product Backlog is transparent, visible and understood.

- The Product Owner **is one person**, not a committee.

Product owner

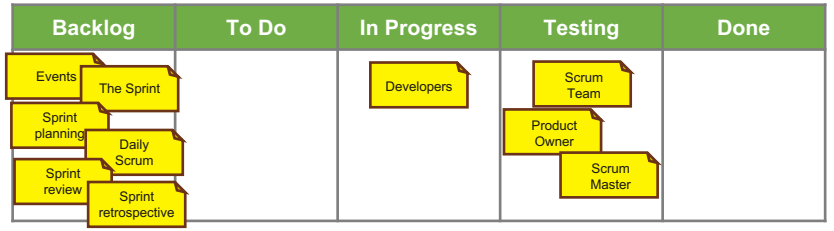
Scrum team – scrum master



- The Scrum Master is **accountable for establishing Scrum**
- Serves the Scrum Team in several ways, including:
  - Coaching the team members in self-management and cross-functionality;
  - Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
  - Causing the removal of impediments to the Scrum Team's progress; and,
  - Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.
- Serves the Product Owner in several ways, including:
  - Helping identify techniques for clear Product Goal and Backlog management;
  - Helping the team understand the need for a concise Product Backlog items;
  - Helping establish empirical product planning for a complex environment; and,
  - Facilitating stakeholder collaboration as requested or needed.

Scrum master

Scrum team – developers



- Developers are the people in the Scrum Team that are committed to **creating any aspect of a usable**

- The specific skills needed by the Developers are often broad and will vary with the domain of work.

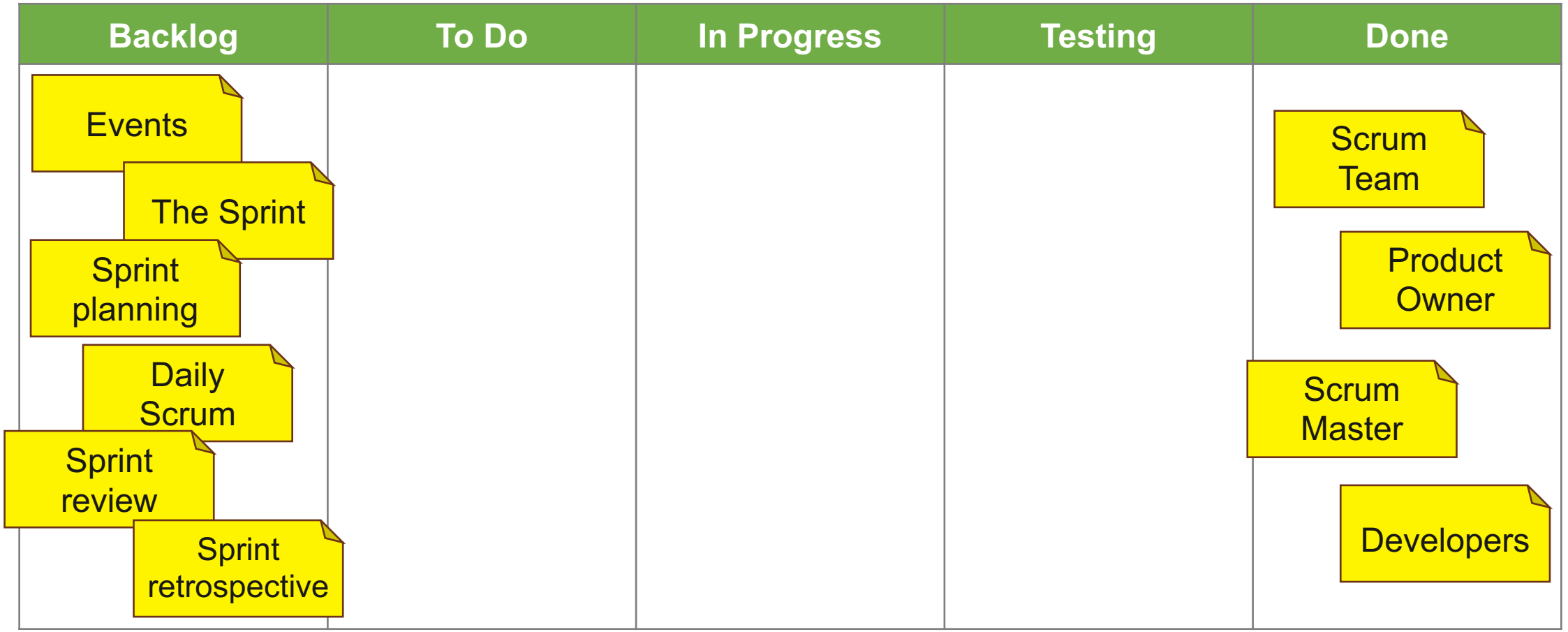
- Developers are always accountable for:
  - Creating a plan for the Sprint, the Sprint Backlog;
  - Instilling quality by adhering to a Definition of Done;
  - Adapting their plan each day toward the Sprint Goal; and,
  - Holding each other accountable as professionals.

Developer

Summary notes

- Scrum Team, only three roles (no more, no less)
  - Product owner, Scrum master and Developers

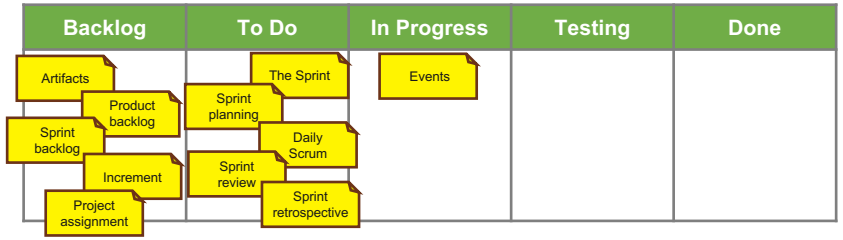
Learn Scrum with Scrum – Sprint #1



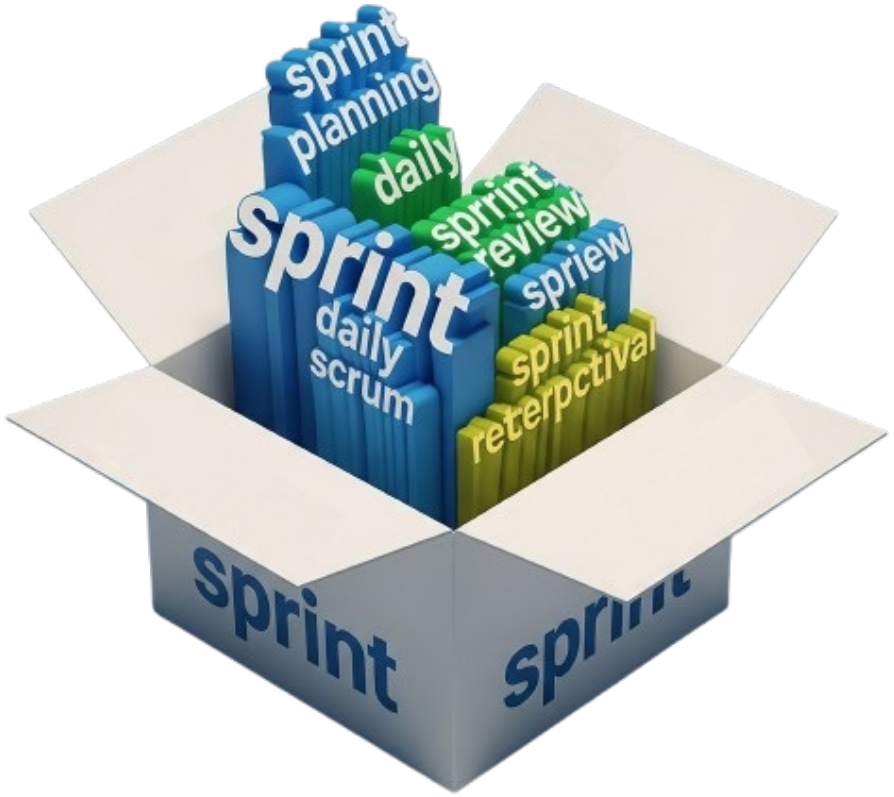
Learn Scrum with Scrum – Sprint #2

Backlog	To Do	In Progress	Testing	Done
Artifacts Product backlog Sprint backlog Increment Project assignment	Events The Sprint Sprint planning Daily Scrum Sprint review Sprint retrospective			

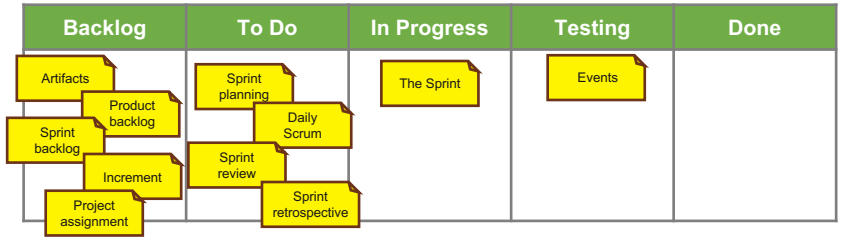
Scrum events



- The Sprint **contains all Scrum events**
- Each event is a **formal opportunity** to inspect and adapt artifacts
- Events ensure **transparency and regularity**
- Skipping events means losing chances to inspect and adapt
- Events **minimize the need** for extra meetings
- Ideally, all events occur **at the same time and place** to reduce complexity

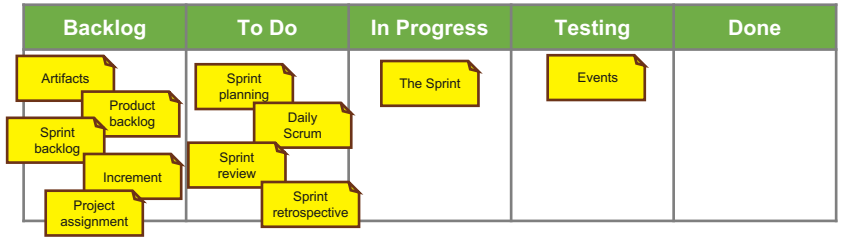


Scrum events – the sprint



- Sprints are the heartbeat of Scrum, where ideas are turned into value.
- They are fixed length events of one month or less to create consistency.
  - A new Sprint starts immediately after the conclusion of the previous Sprint.
- During the Sprint:
  - No changes are made that would endanger the Sprint Goal;
  - Quality does not decrease;
  - The Product Backlog is refined as needed; and,
  - Scope may be clarified and renegotiated with the Product Owner as more is learned.
- A Sprint could be cancelled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint.

Scrum events – the sprint



Responsibilities

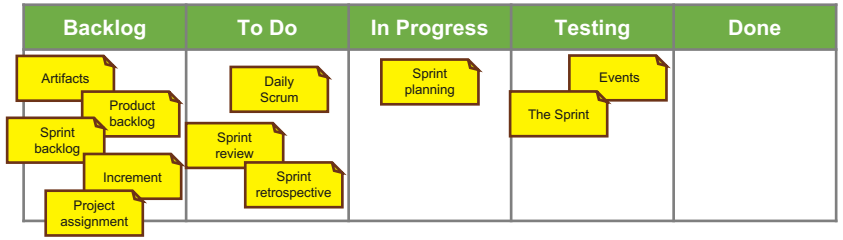
➢ Product Owner

- Ensures that the **Product Backlog is clear, transparent, and reflects current priorities**.
- **Works with stakeholders** to maximize product value.
- Available to **clarify backlog items** during the Sprint.

➢ Scrum Master

- Ensures that the **Scrum framework is understood and followed.**
- Coaches the team and organization on Scrum practices.
- **Removes impediments** to the Scrum Team's progress.
- Helps the Scrum Team focus on **creating high-value Increments**.

Scrum events – sprint planning



- Sprint Planning initiates the Sprint by laying out the work to be performed

It addresses the following topics:

1. Why is this Sprint valuable?

- The Product Owner proposes value improvements, and the Scrum Team sets a Sprint Goal

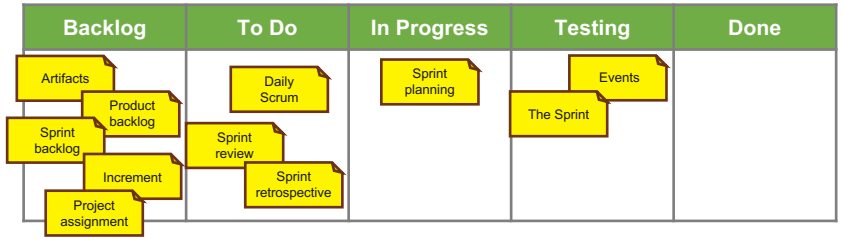
2. What can be Done during this Sprint?

- Developers select items from the Product Backlog to include in the current Sprint.

3. How will the chosen work get done?

- For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done.

Scrum events – sprint planning



Responsibilities

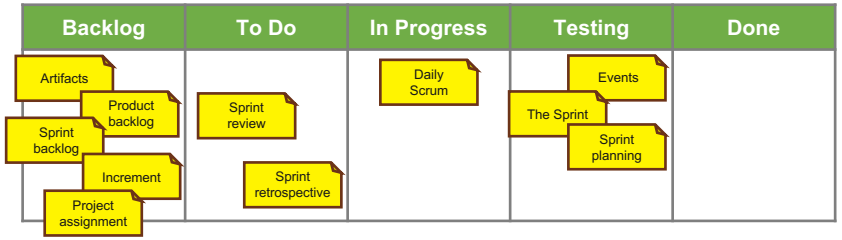
➢ Product Owner

- Presents the Product Goal and **proposes how the product could increase in value** during the Sprint.
- Ensures that **the most valuable Product Backlog Items** are ready for discussion.
- Works **with Developers to define the Sprint Goal** and select items from the Product Backlog.

➢ Scrum Master

- **Ensures the event takes place** and that its purpose is understood.
- Teaches the Scrum Team to keep it within the timebox.
- Facilitates discussion and **collaboration between the Product Owner and Developers**.

Scrum events – daily scrum



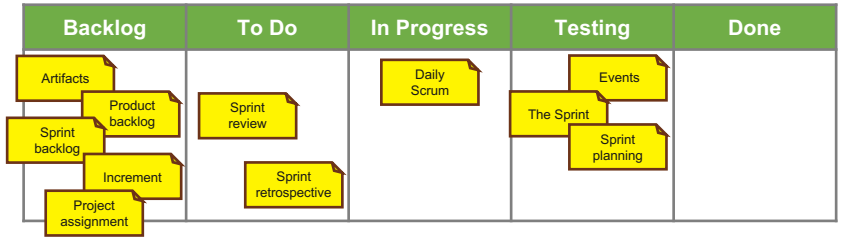
- The Daily Scrum is a 15-minute event for the Developers **synchronize the work** and **plan the next 24 hours** to progress toward the Sprint Goal.

- In the Daily Scrum, team members typically address three questions:
  1. What did I do yesterday to help achieve the Sprint Goal?
  2. What will I do today to help achieve the Sprint Goal?
  3. Are there any impediments blocking my progress?

- Improve communications, **identify impediments**, promote **quick decision-making**
  - Consequently, eliminate the need for other meetings

Scrum events – daily scrum

Responsibilities



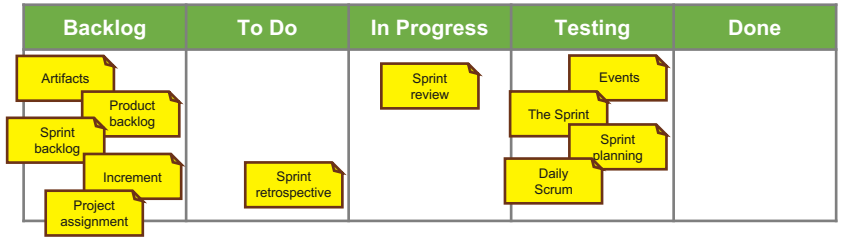
➢ Product Owner

- Usually does not participate unless they are actively working on Sprint Backlog items.
- If present, listens to progress and adapts Product Backlog if needed outside of the Daily Scrum.
- The only time a Product Owner participates as a full-fledged member of the Daily Scrum is if they are **actively working on items in the Sprint Backlog.**
  - In this case, they would be acting as a developer

➢ Scrum Master

- Ensures the Developers hold the event and understand its purpose.
- Ensures that the **15-minute timebox is not exceeded**.
- Coaches them to keep it focused on progress toward the Sprint Goal and on adjusting the plan for the next 24 hours.

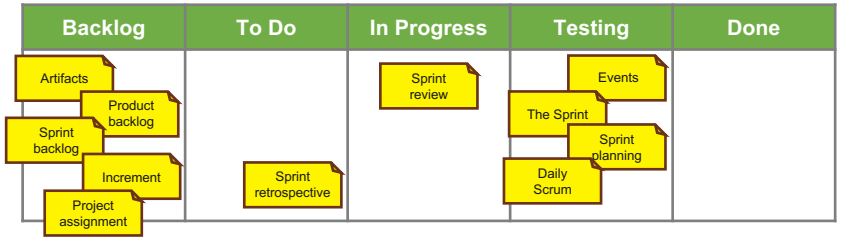
Scrum events – sprint review



- It is held at the **end of the Sprint**
  - To inspect the increment and determine future adaptations
- The Scrum Team presents the results of their work to key stakeholders
  - And progress toward the Product Goal is discussed
- The Sprint Review is a working session
  - The Scrum Team should avoid limiting it to a presentation
- The Sprint Review is the second to last event of the Sprint

Scrum events – sprint review

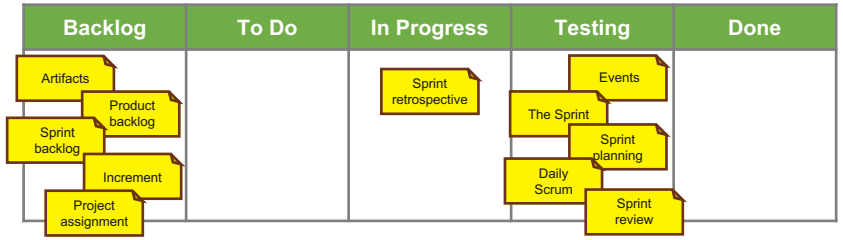
Responsibilities



- Product Owner
  - Explains what **Product Backlog items were "Done"** and what has not been completed.
  - Discusses the Product Backlog as it stands.
  - Proposes likely next steps to increase product value.
  - Works with **stakeholders** and Scrum Team to **decide on future adaptations.**

- Scrum Master
  - Ensures the event happens and that participants understand its purpose.
  - Keeps it within the timebox.
  - Facilitates **collaboration between Scrum Team and stakeholders**.

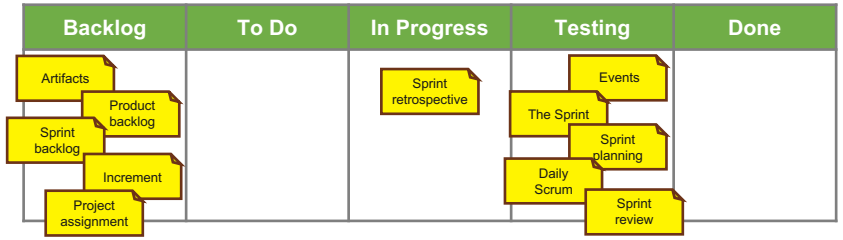
Scrum events – retrospective



- The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.
- The Scrum Team identifies the most helpful changes to improve its effectiveness.
  - The most impactful improvements are addressed as soon as possible.
  - They may even be added to the Sprint Backlog for the next Sprint.
- The Sprint Retrospective concludes the Sprint.

Scrum events – retrospective

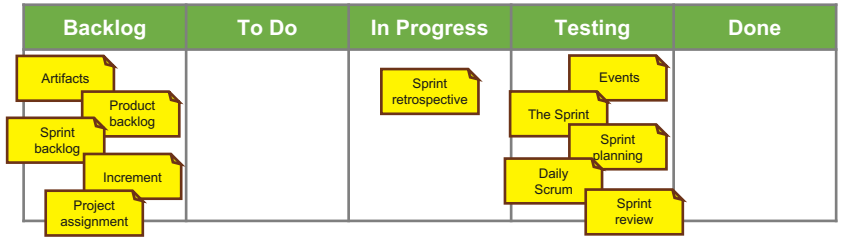
Responsibilities



- Product Owner
  - Participates as a Scrum Team member.
  - Collaborates on **identifying improvements for processes, tools, relationships, and work quality.**

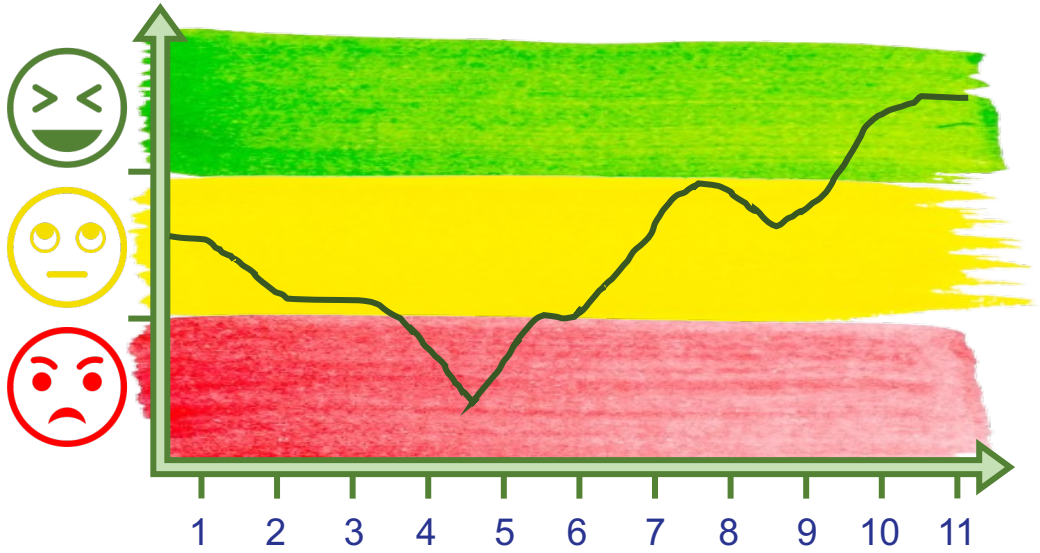
- Scrum Master
  - Ensures the event occurs and that its purpose is clear.
  - Encourages the team to inspect itself and **create a plan for improvements**.
  - Promotes Scrum practices that improve effectiveness.

Scrum events – retrospective

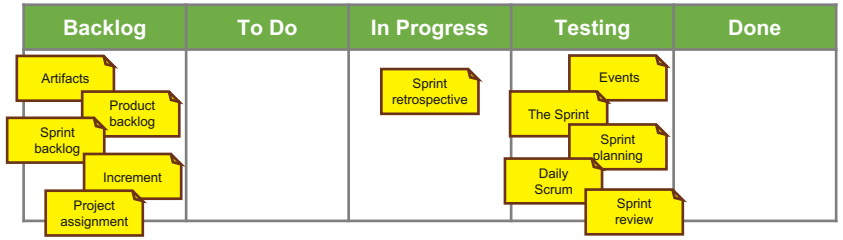


Happiness

- Discuss what **went well**, what **could be improved**, and **quick wins**
- Happy teams are more productive and Creative
- Measuring happiness helps to **identify changes that boost morale** and efficiency



Scrum events – retrospective



Happiness

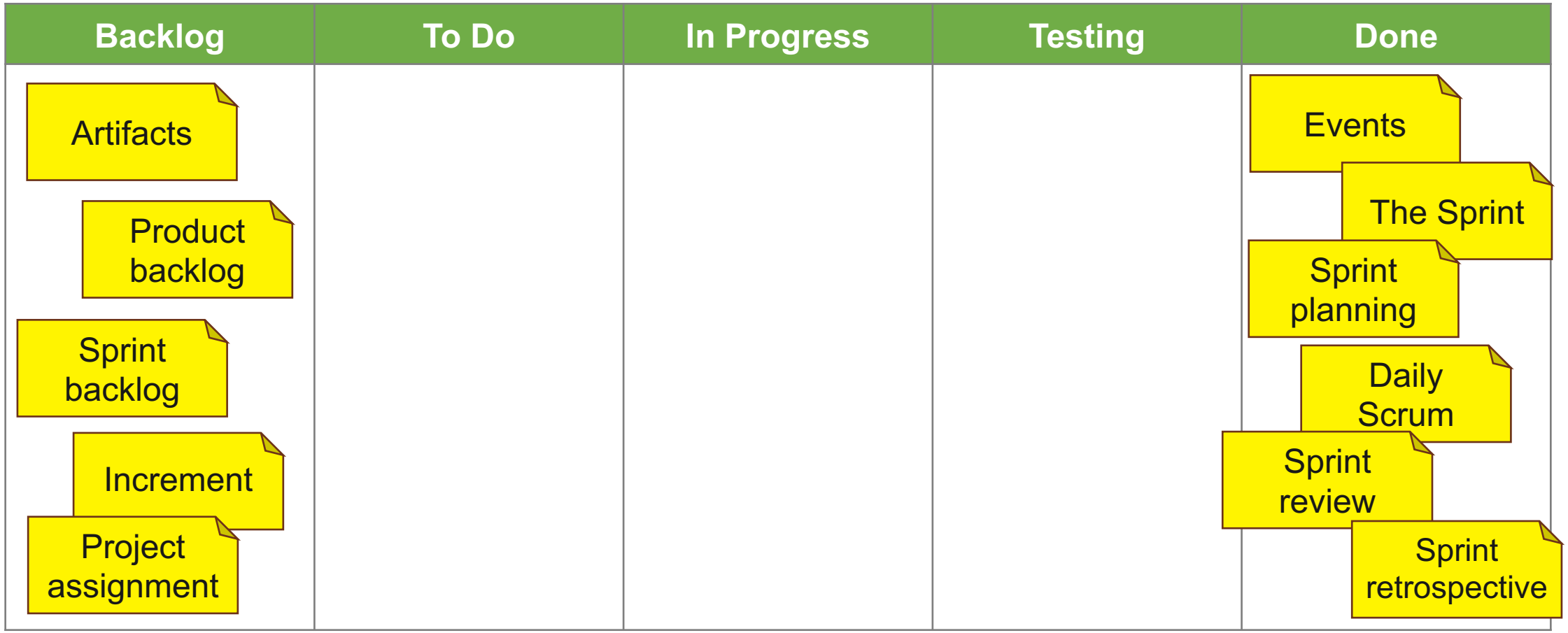
At the end of each Sprint each person on the team answers just a few questions:

1. On a scale from 1 to 5, how do you feel about your role in the company?
2. On the same scale, how do you feel about the company as a whole?
3. Why do you feel that way?
4. What one thing would make you happier in the next Sprint?

Summary notes

- Scrum Team, only three roles (no more, no less)
  - Product owner, Scrum master and Developers
- Scrum Events, five in total
  - The sprint, sprint planning, daily scrum, sprint review and sprint retrospective

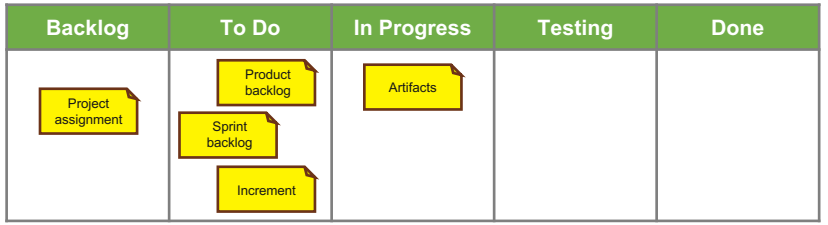
Learn Scrum with Scrum – Sprint #2



Learn Scrum with Scrum – Sprint #3

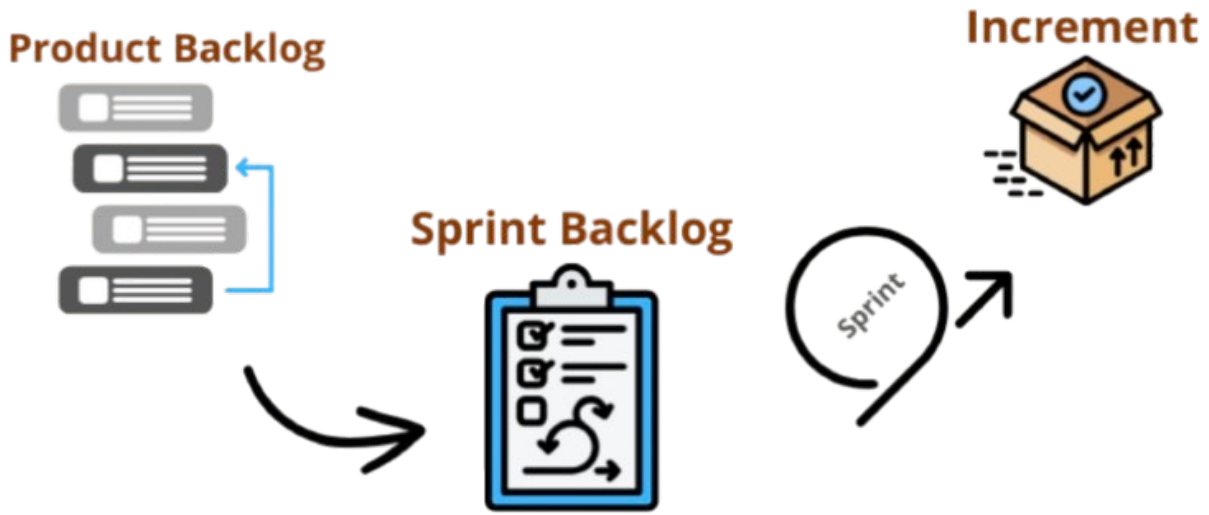
Backlog	To Do	In Progress	Testing	Done
Project assignment	Artifacts Product backlog Sprint backlog Increment			

Scrum artifacts

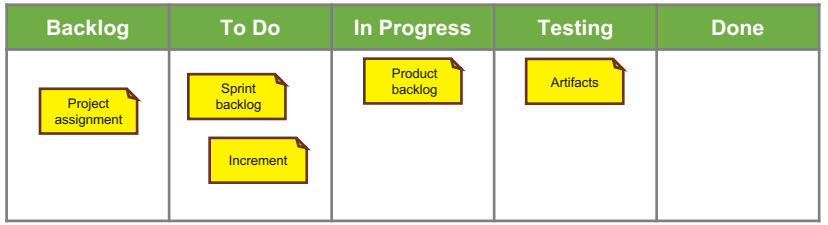


- Scrum's artifacts represent work or value.
  - They are designed to maximize transparency of key information.
  - Everyone inspecting them has the same basis for adaptation.

- Each artifact provides measurable information to enhance transparency and focus.
  - Product Backlog → **Product Goal**
  - Sprint Backlog → **Sprint Goal**
  - Increment → **Definition of Done**



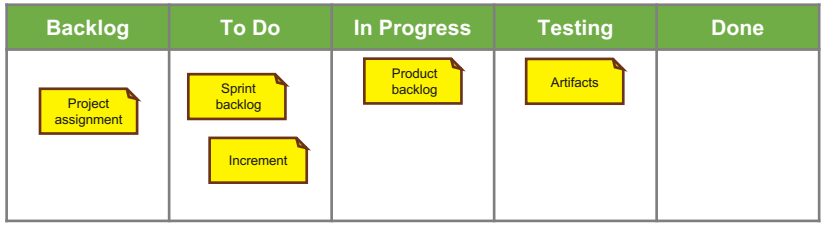
Scrum artifacts – product backlog



- The Product Backlog is an emergent, ordered list of what is needed to improve the product.
  - It is the single source of work undertaken by the Scrum Team.
- Product Backlog items that can be Done by the Scrum Team
  - Within one Sprint are deemed ready for selection in a Sprint Planning event.
  - They usually acquire this degree of transparency after refining activities.
- **Commitment: Product Goal**

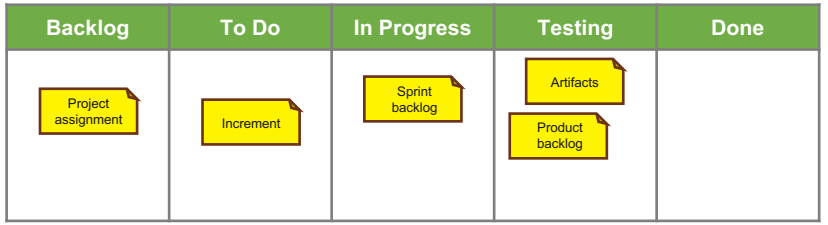
A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users or customers. A product could be a service, a physical product, or something more abstract.

Scrum artifacts – product backlog



- The Developers who will be doing the work are responsible for the sizing.
  - The Product Owner may influence the Developers by helping them understand and select trade-offs.
- Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items.
  - This is an ongoing activity to add details, such as a description, order, and size. Attributes often vary with the domain of work.
- The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against.
  - The Product Goal is in the Product Backlog.
  - The rest of the Product Backlog emerges to define “what” will fulfill the Product Goal.

Scrum artifacts – sprint backlog

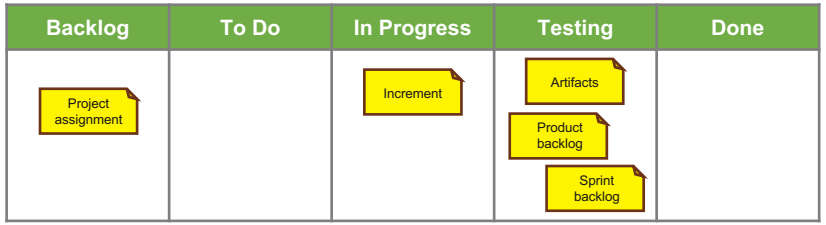


- The Sprint Backlog is composed of the Sprint Goal (why)
- The set of Product Backlog items selected for the Sprint (what)
- The actionable plan for delivering the Increment (how).

- The Sprint Backlog is a plan by and for the Developers.
  - It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal.
  - Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned.
  - It should have enough detail that they can inspect their progress in the Daily Scrum.

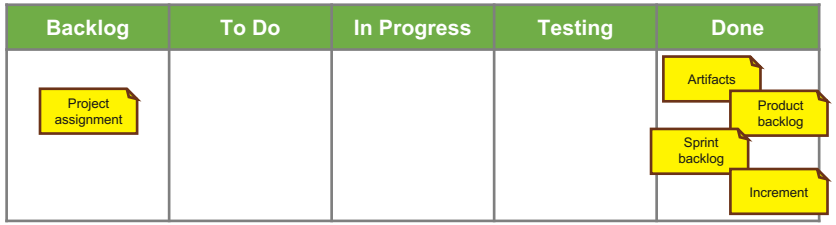
- **Commitment: Sprint Goal**

Scrum artifacts – increment



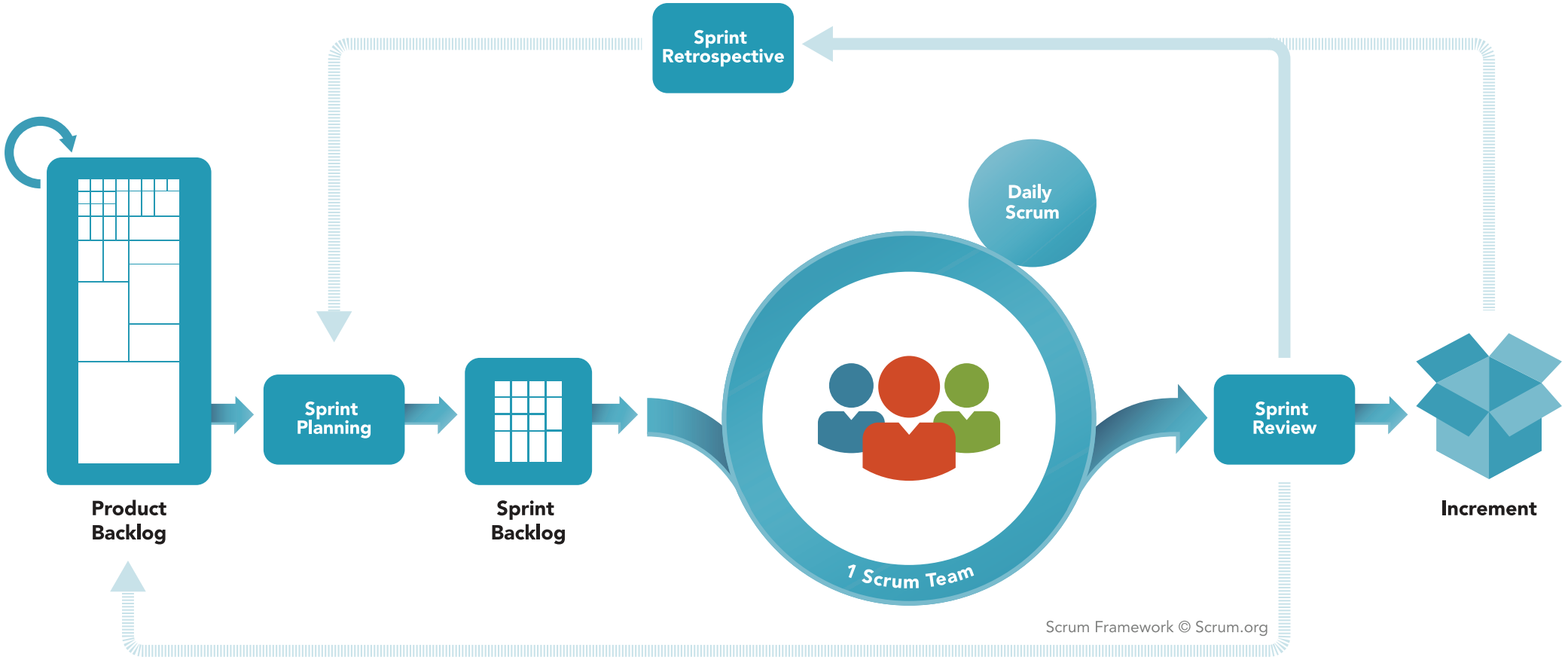
- An Increment is a concrete stepping stone toward the Product Goal.
  - Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together.
- Multiple Increments may be created within a Sprint.
  - The sum of the Increments is presented at the Sprint Review thus supporting empiricism.
  - The Sprint Review should never be considered a gate to releasing value.
- Work cannot be considered part of an Increment unless it meets the Definition of Done.
  - The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.
- **Commitment: Definition of Done**

Summary notes



- Scrum Team, only three roles (no more, no less)
  - Product owner, Scrum master and Developers
- Scrum Events, five in total
  - The sprint, sprint planning, daily scrum, sprint review and sprint retrospective
- Scrum Artifacts, also three
  - Product backlog, sprint backlog and increment
- It is just a 3-5-3!
  - But requires discipline

Scrum – overview



Project assignment

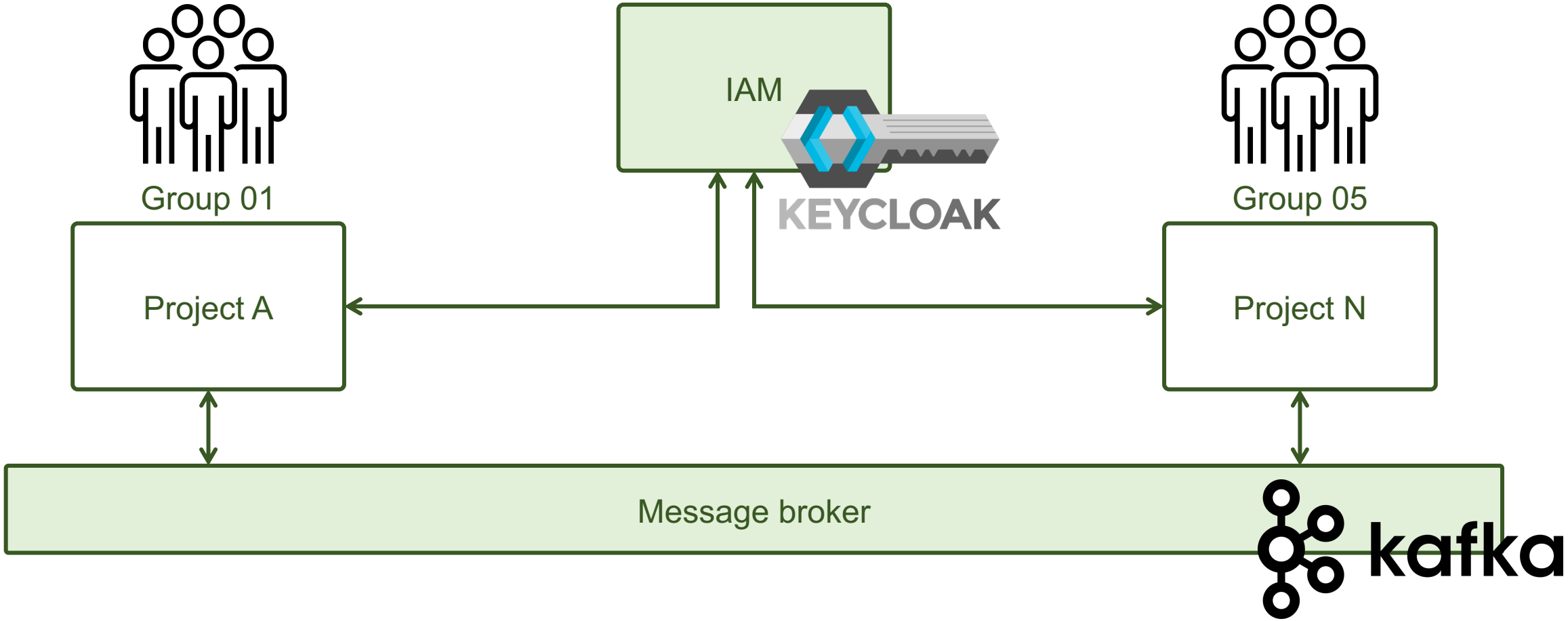
Project goal

- Solve a unique problem: A modular and event-driven enterprise operations management

- Project divided into smaller projects (components)
  - Each group develops a **standalone component**
  - Components can run independently but may depend on others
    - **Message queue system** for communication & data sharing

- Environment is **volatile**
  - Client needs are expected to evolve – requirements will change
  - Use **adaptive project management** to handle evolving needs

Overview


IAM
KEYCLOAK
Group 01
Group 05
Project A
Project N
Message broker
kafka

Team/groups

- The project is formed by groups of five students
  - The group should elect the Product Owner and the Scrum Master
  - All members will be developers
- Roles are collaborative, not hierarchical – **decisions made as a team**
- Other members can represent specific domains in the project
  - Example: one person responsible for deployments

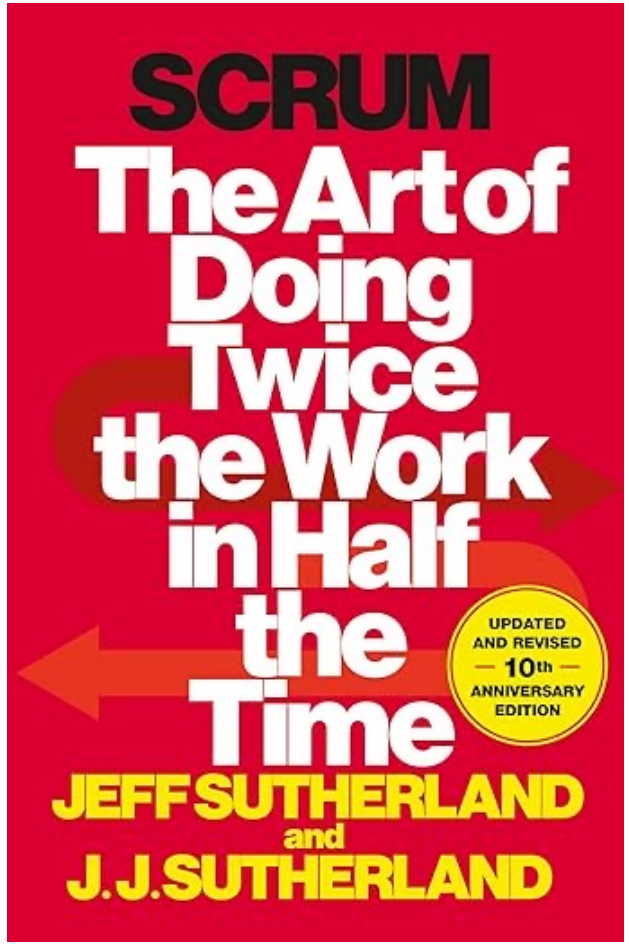
Deliverables and sprints

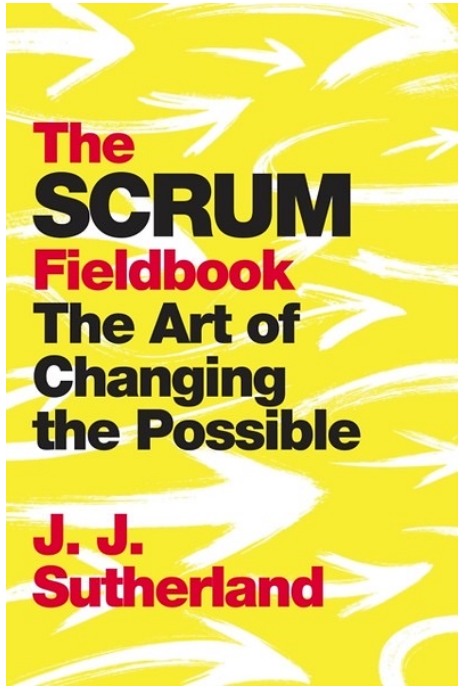
Sprint	Objectives
1	Establish the technical backlog, implement the continuous integration process, define core user stories, and design the component architecture.
2	Deliver the first MVP, incorporating initial user stories
3	Release Version 2.x.x, which includes new requirements, an initial testing strategy, a process for releasing new features, and full integration into our ecosystem.
4	Release software version 3.x.x, including tests for new features, the introduction of Infrastructure as Code (IaC), and the first version of Service Level Objectives (SLOs), while addressing new client requirements.
5	Release software version 4.x.x, adding system observability, enhancing Service Level Objectives, and integrating new client requirements.
6	Release software version 5.x.x, create the environment for post-mortems, and apply final refinements to the system.
7	Final presentations.

Mandatory technologies

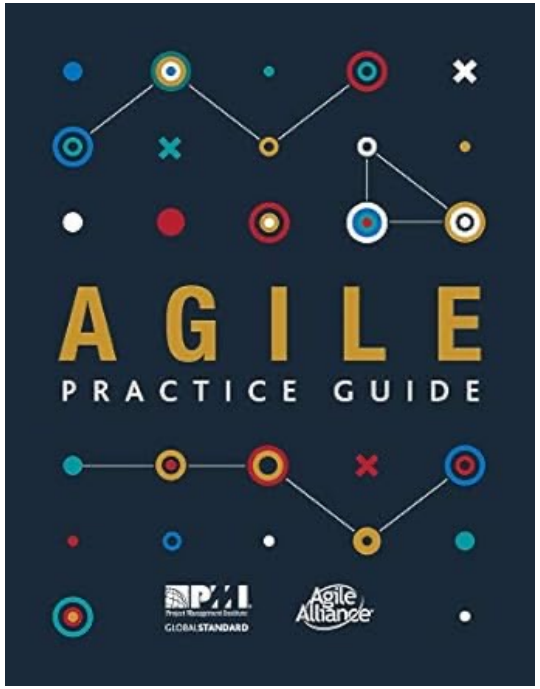
- Backend
  - Spring boot
- Frontend
  - ReactJS
- Communication
  - Between backend and frontend: RESTfull API
  - Between projects: Message queue with Apache Kafka
- Identity Provider
  - Keycloak managed by teaching staff

References


SCRUM
The Art of Doing Twice the Work in Half the Time
UPDATED AND REVISED
10th
ANNIVERSARY EDITION
JEFF SUTHERLAND
and
J.J. SUTHERLAND


The
SCRUM
Fieldbook
The Art of Changing the Possible
J. J. Sutherland


SCRUM
A Gestão Ágil de Projetos
JOÃO PAULO PINTO
CHRISTIANE TSCHARF
FCA


AGILE
PRACTICE GUIDE
PMI
GLOBAL STANDARD
Agile Alliance