



Fall 2025

L03 Scrum

CS 1530 Software Engineering

Nadine von Frankenberg



University of
Pittsburgh

School of Computing
and Information

University of Pittsburgh
School of Computing and Information
sci.pitt.edu

Copyright

- These slides are intended for use by students in CS 1530 at the University of Pittsburgh only and no one else. They are offered free of charge and must not be sold or shared in any manner. Distribution to individuals other than registered students is strictly prohibited, as is their publication on the internet.
 - All materials presented in this course are protected by copyright and have been duplicated solely for the educational purposes of the university in accordance with the granted license. Selling, modifying, reproducing, or sharing any portion of this material with others is prohibited. If you receive these materials in electronic format, you are permitted to print them solely for personal study and research purposes.
 - Please be aware that failure to adhere to these guidelines could result in legal action for copyright infringement and/or disciplinary measures imposed by the university. Your compliance is greatly appreciated.
- Material from these notes is obtained from various sources, including, but not limited to, the following:
 - Bruegge, & Dutoit. Object-oriented software engineering. using UML, patterns, and Java. Pearson, 2009.
 - Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns. Pearson, 1994.
 - Sommerville, Ian. "Software Engineering" Pearson. 2011.
 - <http://scrum.org/>

Learning goals

- You have a better overview of the software development lifecycle
- You understand the roles, artifacts, events, and meetings of Scrum

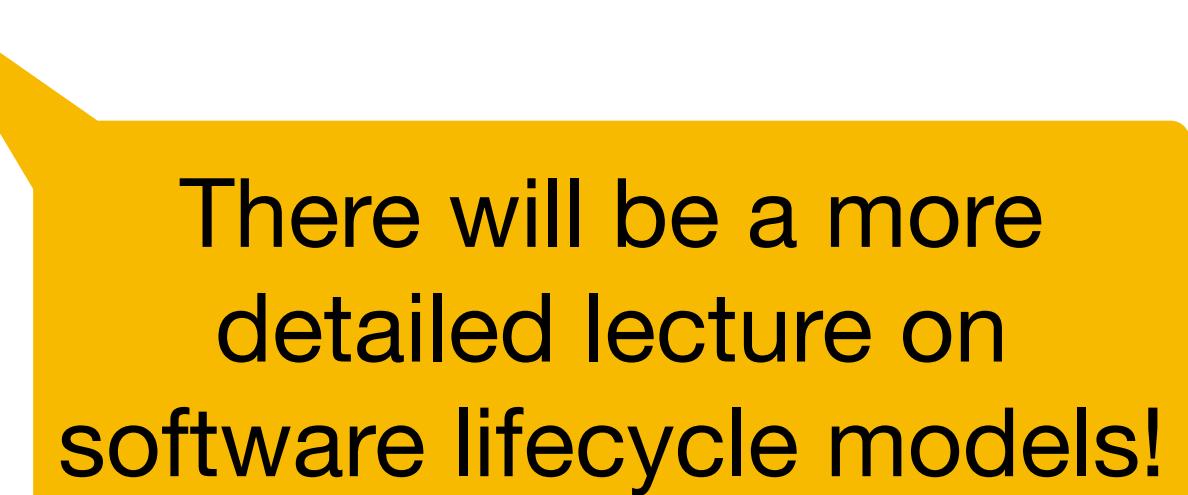
Today's roadmap

→ Software lifecycle overview

- Scrum

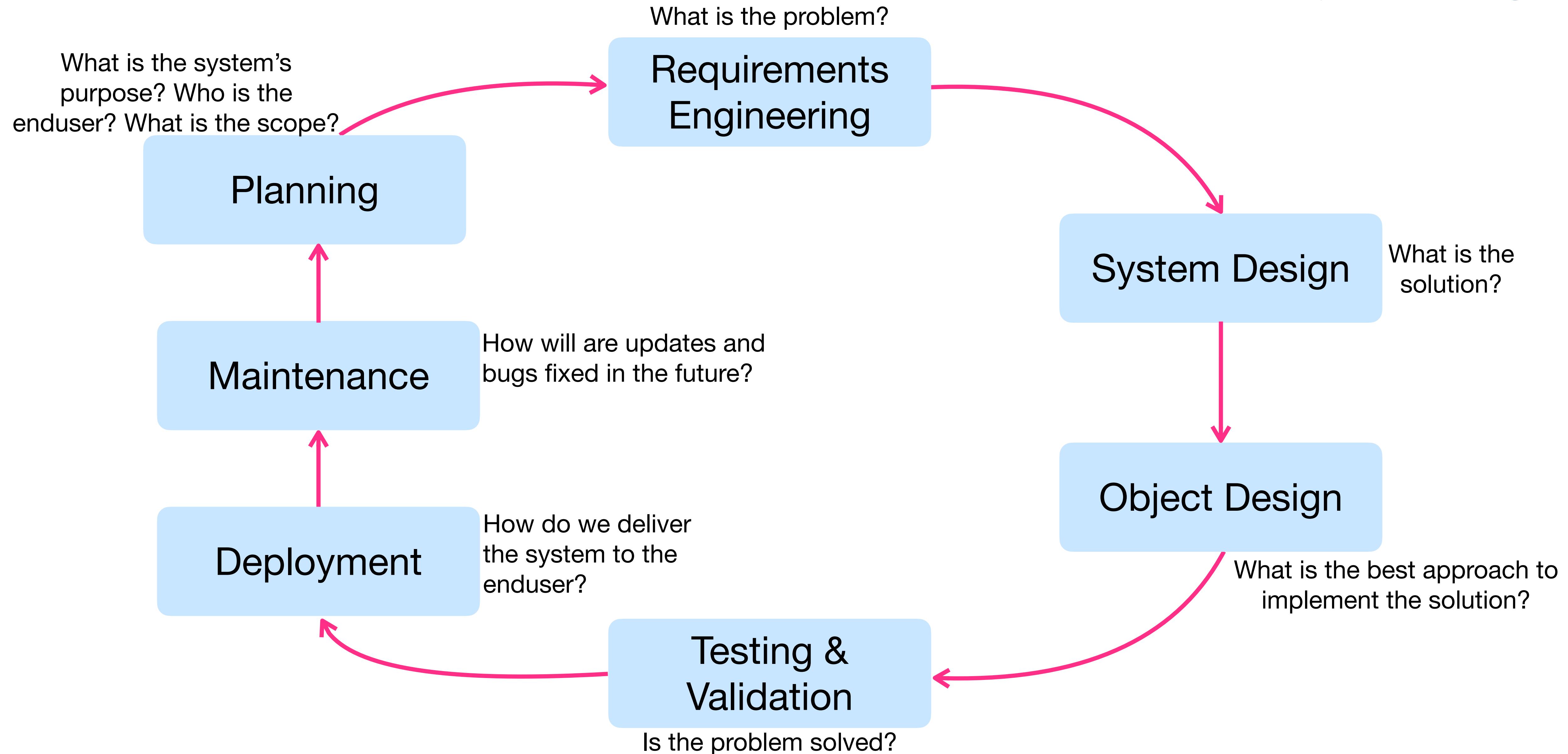
[Recap] SDLC

- A **software (development) life cycle (SDLC)** describes the **process of how software is developed**
 - **Set of activities** and their relationships to each other to support the development of a software system
 - Activities include, e.g., requirements elicitation, requirements analysis, system design, implementation, testing & validation, configuration management, delivery, maintenance
- Defining a SDLC is necessary:
 - Establish a shared understanding (communication)
 - Creates a plan (releases / delivery)
 - Incorporate all aspects (testing, documentation, maintenance)
 - ...



There will be a more detailed lecture on software lifecycle models!

Overview of the SDLC



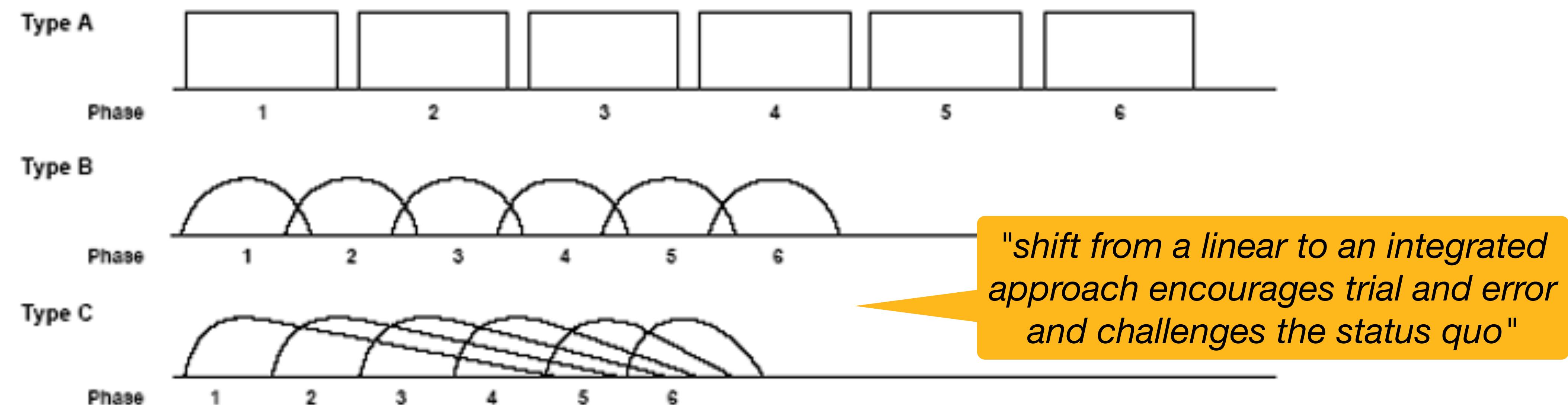
Today's roadmap

- Software lifecycle overview
- Scrum

History of agile models – Scrum

- In 1986, Takeuchi and Nonaka published "*The New New Product Development Game*" in the *Harvard Business Review*
- Equates creative processes (e.g., Honda, 3M, HP, ..) to the sport of rugby
- Describes a scalable, flexible, cross-functional, self-organized team-based approach to product development

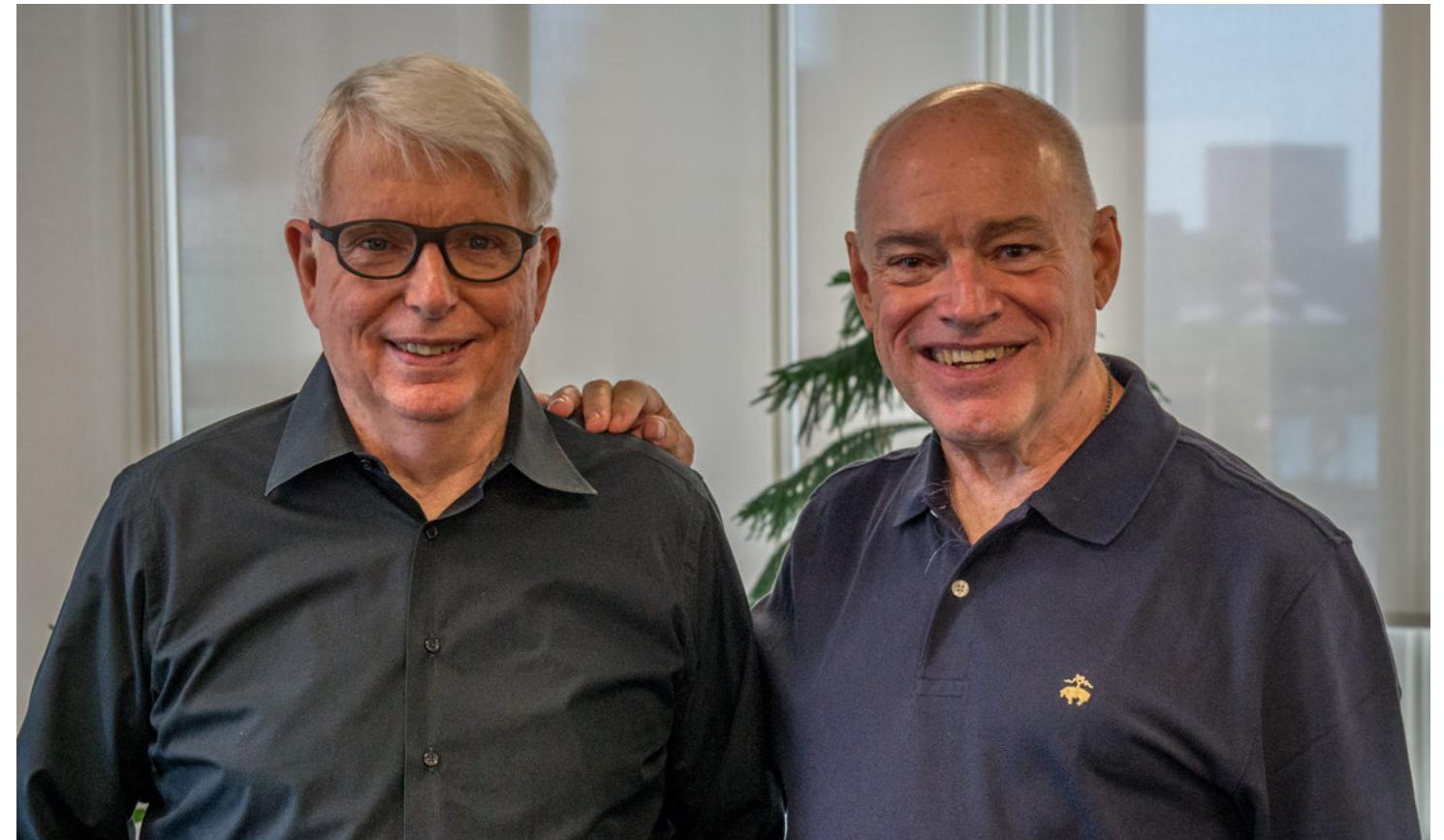
Sequential (A) vs. overlapping (B and C) phases of development



<https://hbr.org/1986/01/the-new-new-product-development-game>

History of agile models – Scrum

- In 1986, Takeuchi and Nonaka published "The New New Product Development Game" in the *Harvard Business Review*
- 1987 "SCRUM development process"
- 1995 **Jeff Sutherland and Ken Schwaber** analyzed common software development processes
 - "The Scrum development process" at OOPSLA
 - "*Systems development [...] is an unpredictable and complicated process that can only roughly be described as an overall progression*"
- 2001 The Agile Manifesto
- 2002 Scrum Alliance (Mike Cohn, Esther Derby, Ken Schwaber)



Jeff Sutherland

Ken Schwaber

Scrum

- Framework for developing and sustaining complex products
- Scrum defines roles, events, artifacts, and the rules that bind them together
- "The Scrum Guide" written by Ken Schwaber and Jeff Sutherland
 - First version in 2010, latest updated version 2020

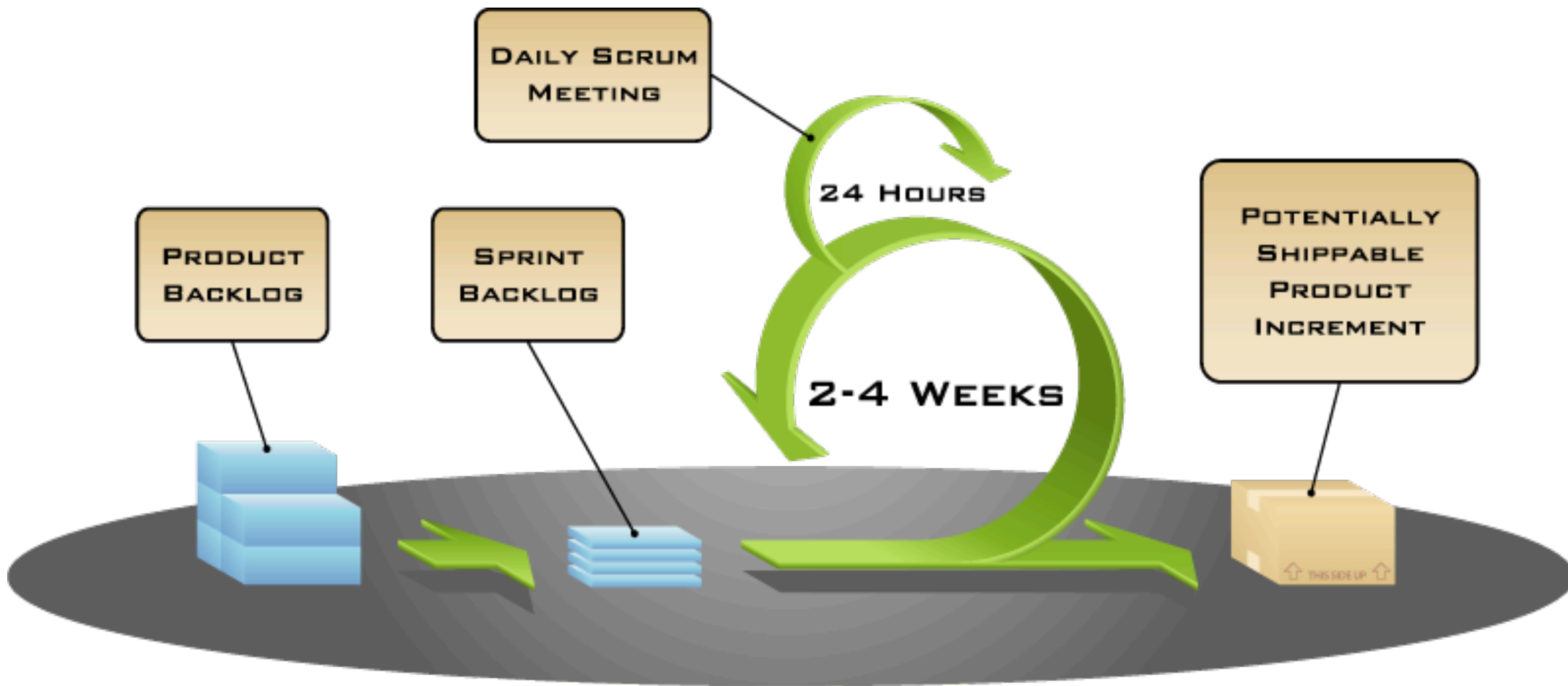


[Rugby] **Scrum** is a **way to restart the game after an interruption**: Players pack closely together with their heads down to gain possession of the ball
→ Requires coordination, joint effort, strategic planning

Overview: Scrum as an agile model

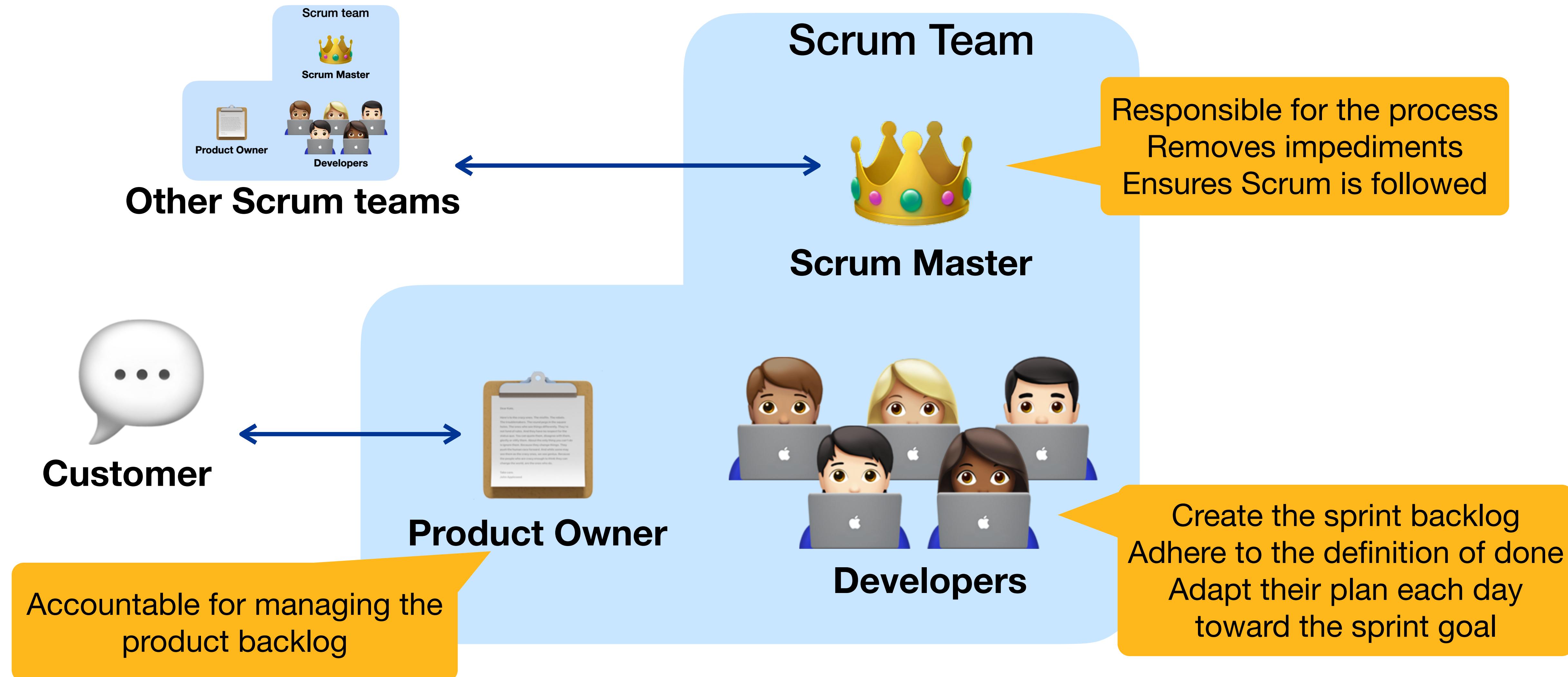
- Short, iterative cycles (**Sprints**), typically 2-4 weeks long
- Emphasizes collaboration, functioning software, and flexibility to changing requirements
- Roles: **Scrum Master, Product Owner, Development Team**
- Promotes transparency, inspection, and adaptation throughout the process

Scrum



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

The Scrum Team - Roles



The Scrum Team

- Consists of
 - One Scrum Master
 - One Product Owner
 - Developers
- No sub-teams or hierarchies
- Scrum Teams are cross-functional
- Max. 10 team members
- Accountable for creating a valuable Increment each Sprint



The Scrum Team - Product Owner

- "Owns" the product
- Accountable for maximizing the product's value
- Tasks
 - Manages the Product Backlog
 - Develops and communicates the product goal
 - Creates and communicates Product Backlog Items
 - Orders Product Backlog Items
 - Ensures that the Product Backlog is transparent and understood
- Can represent the needs of stakeholders in the Product Backlog



The Scrum Team - Scrum Master

- Accountable for establishing Scrum
- Serves the Scrum Team
 - Coaches team members
 - Removes impediments
 - Ensures that all Scrum events take place
- Serves the Product Owner
 - Define Product Goal, Product Backlog definition
 - Establish empirical product planning
 - Facilitate stakeholder collaboration as needed

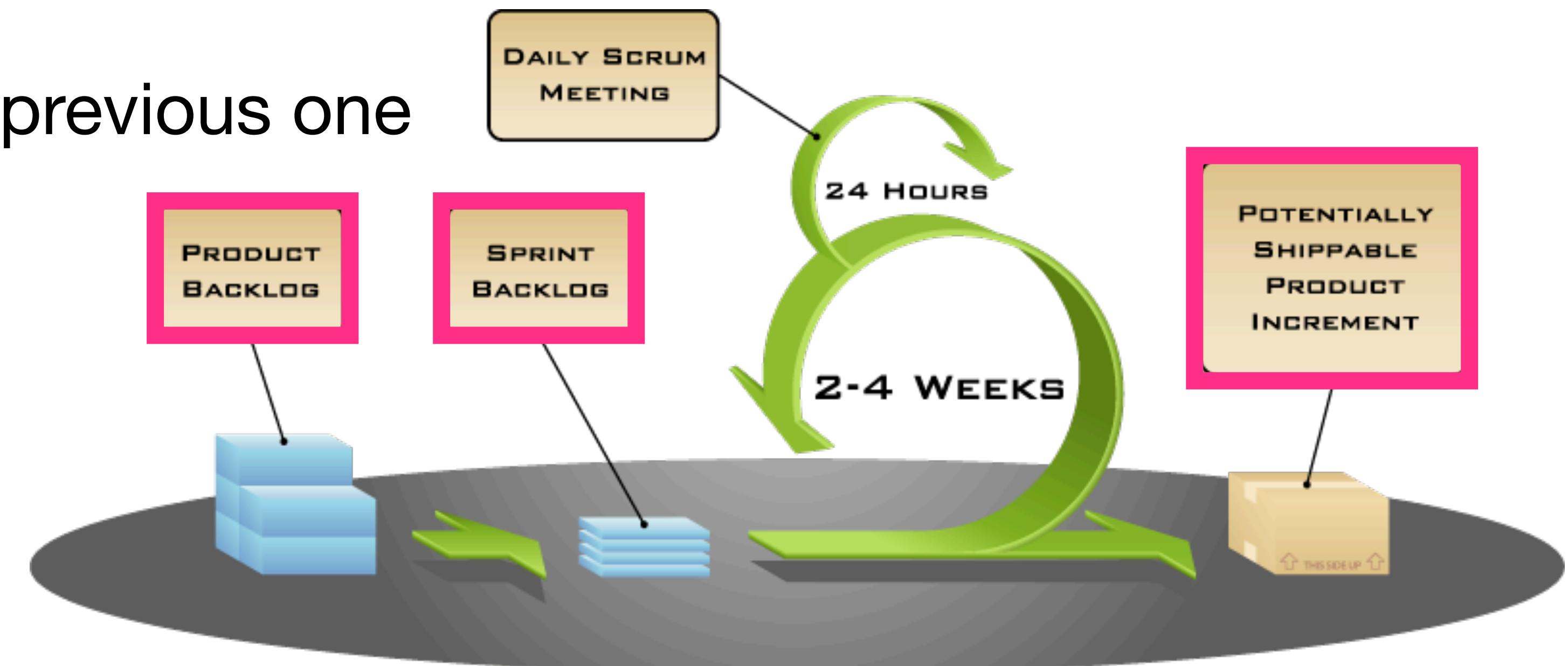


The Scrum Team - Developers

- Committed to creating any aspect of a usable increment each Sprint
- Skills often vary by domain
- Developers are accountable for
 - Creating the Sprint Backlog
 - Adhere to a Definition of Done
 - Adapt their plan daily towards the Sprint Goal

Artifacts in Scrum represent work or value

- Designed to maximize transparency of key information
- **Product backlog**: list of requirements for the whole product
- **Sprint backlog**: list of requirements to be completed in one iteration (sprint)
- **Increment**: sum of all completed work in a Sprint, usable, "potentially shippable" piece of the product, must meet the DoD, each increment builds on the previous one release is optional



Requirements in Scrum

- All **requirements** are collected in the product backlog
- The product owner elicits requirements from external stakeholders & prioritizes them
- Typical requirements in Scrum
 - "Technical"/functional requirements
 - User stories
 - Scenarios
 - Use cases



More on requirements next lecture

[Example] User Story in Scrum

- Sentence that describes what the user can do
- Often written on a "card"
- Users describe roles of the system
- Defines a structure and simplifies the prioritization process
- Facilitates categorization according to user roles

As <role>,
I can/want/need <feature>
so that <reason>

"I"-perspective: developer
can self-identify with the
given user role

[Example]

As a student, I want to sign up
for a course so that I can
access course materials and
participate in lectures.

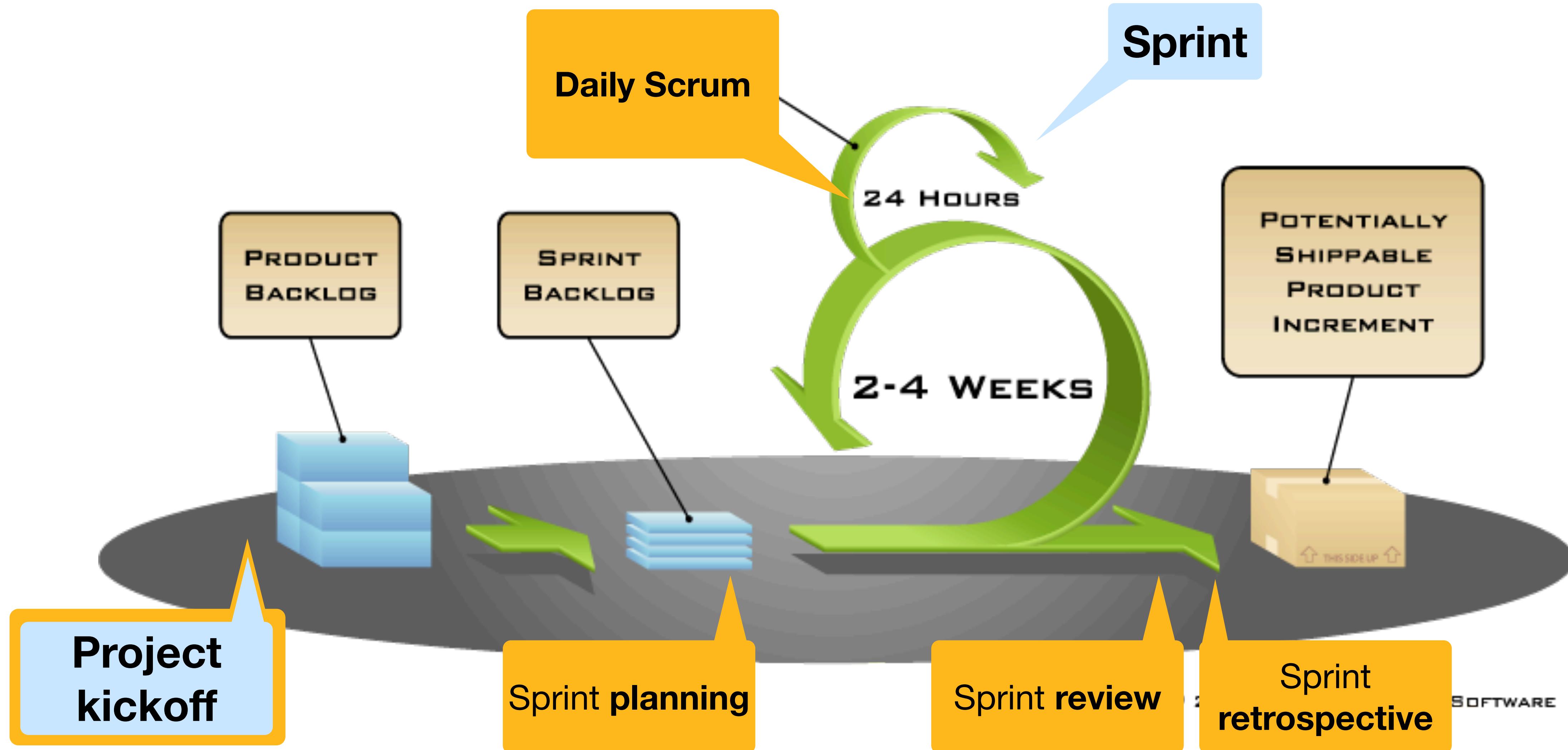
Product backlog

- Ordered list of requirements (what is needed to improve the whole product)
 - E.g., Epics, User Stories, Scenarios
- "*Single source of work undertaken by the Scrum Team*"
- ***Product Backlog refinement*** is the process of reviewing, updating, and prioritizing items in the product backlog
 - Goal: Ensure clarity, relevance, and value
 - Involves breaking down larger tasks, estimating efforts, and reordering items to align with the project's evolving goals and requirements
- Commits to the **product goal**
 - The product goal is the long-term objective for the product, provides a target for the Scrum team to work towards

[Example] Product Backlog

- User account creation: As a player, I want to create a personal account, so I can save my progress and scores.
- Quiz category selection: As a player, I want to select a quiz category before I start the quiz, so I can choose a topic that interest me.
- Difficulty levels: As a player, I want to choose different difficulty levels, so I can challenge myself appropriately.
- Answering Questions: As a player, I want to answer multiple-choice questions, so I can engage with the quiz format that is familiar and straightforward.
- Time-Limited questions: As a player, I want each question to be time-limited (max. 1min per question), so that the game is more challenging and fair.
- Score tracking: As a player, I want the game to track my scores, so I can see how well I am doing.
- Leaderboard Functionality: As a player, I want to view a leaderboard, so I can compare my scores with other players.
- Accessibility: As a player with visual impairments, I need accessibility features (text-to-speech), so I can fully participate in the quizzes.

Two main events & five meeting types in Scrum

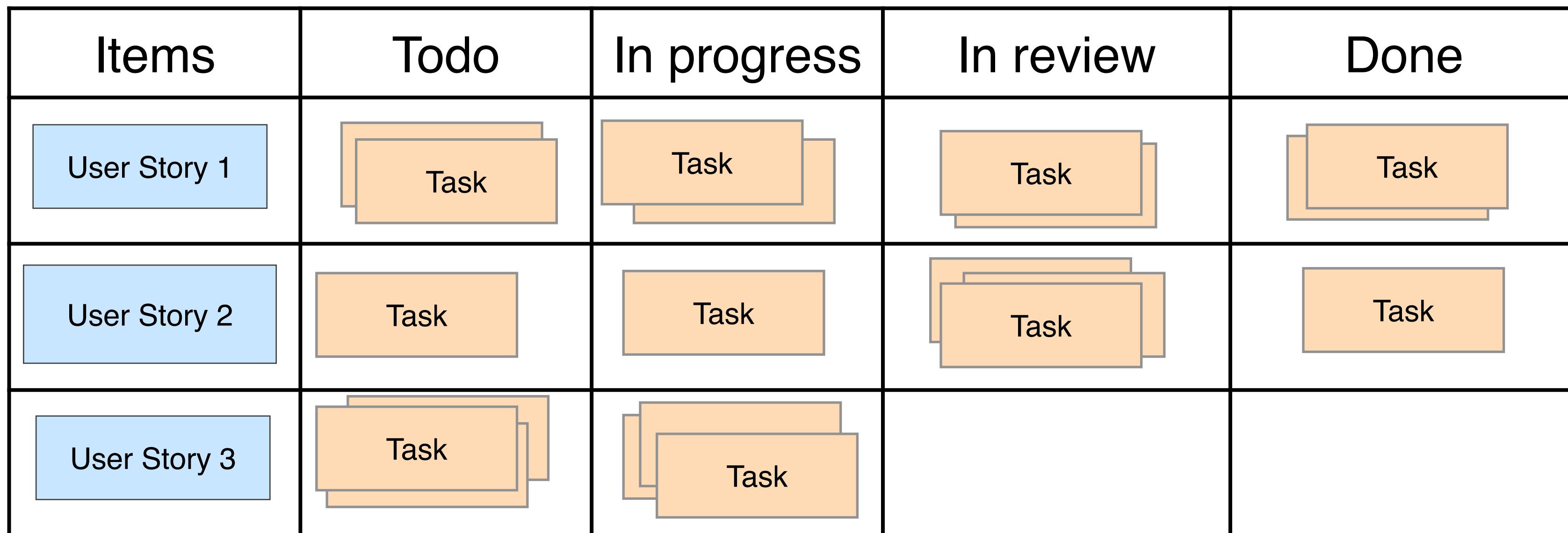


Main events in Scrum

- **Project kickoff:** Define the product backlog
- **Sprint** "*turns ideas into value*"
 - Fixed length of **one month or less** to create consistency
 - A new sprint starts immediately after concluding the previous sprint
 - A sprint commits to the **sprint goal**

The Sprint

- The development team
 - Realizes the items in the sprint backlog
 - Uses, e.g., a task board to visualize the status of these items
- The Scrum Master visualizes the progress



During a Sprint ...

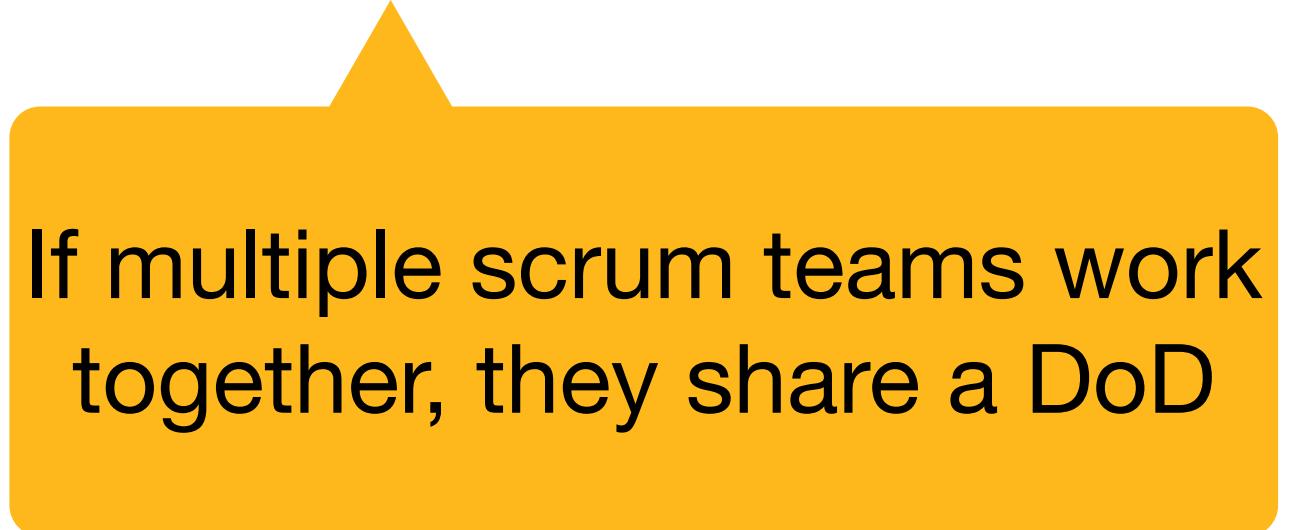
- Start of the Sprint: Sprint Planning meeting [max. 8h]
 - Create Sprint Backlog
- Every day: Daily Scrum [max. 15min]
 - Share status, impediments and promises
- End of each Sprint: Sprint Review [max. 4h]
 - Review the realized Sprint Backlog Items with Product Owner
- After the Sprint: Sprint Retrospective [max. 3h]
 - Inspect the previous sprint
 - Create a plan for improvement (individuals, interactions, processes, tools, DoD) that will set the basis for the next Sprint

Sprint Planning

- Initiates the Sprint
- Entire Scrum Team
- Addresses three questions:
 - Why is this Sprint valuable?
 - What can be Done this Sprint?
 - How will the chosen work get done?
- Often includes activities to estimate the size of a Backlog Item
 - E.g., Story points
 - T-Shirt Sizes

Definition of Done (DoD)

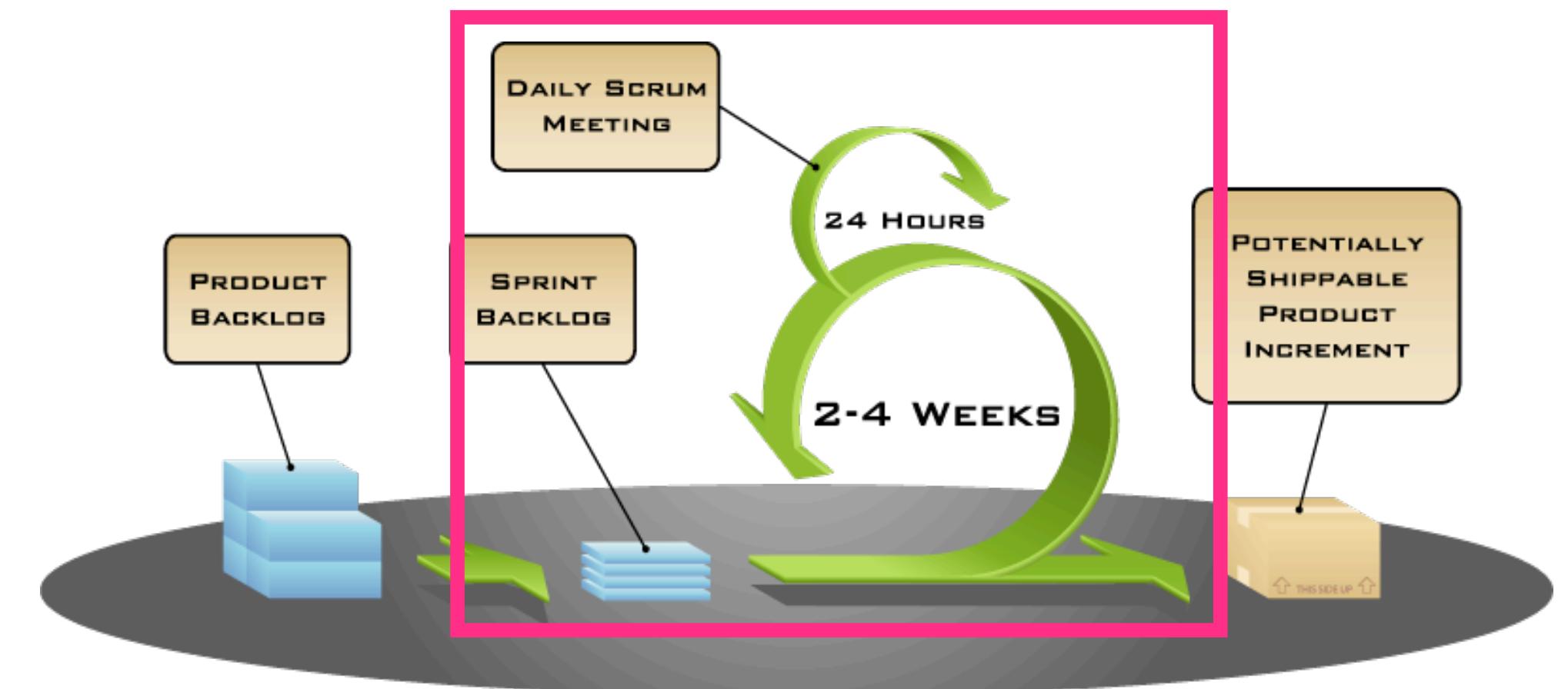
- Formal description of the state of the Increment
- Describes when the Increment is "done" (meets quality requirements)
- Whenever a Product Backlog Item meets the DoD, a new Increment is created
- Creates a shared understanding of what work needs to be completed
- An Increment cannot be released if the DoD is not met
- The Scrum Team follows one DoD



If multiple scrum teams work together, they share a DoD

The Sprint

- During the Sprint:
 - No changes are made that would endanger the Sprint Goal
 - Maintain quality
 - Refine the Product Backlog as needed
 - Scope may be clarified and renegotiated with the Product Owner, based on insights
- A Sprint may be cancelled if the Sprint Goal is no longer relevant
 - The Product Owner has the sole authority to cancel a Sprint


COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Daily Scrum

- Time-boxed event for 15 min
- Attended by the Developer Team
- Developer Team sets structure and techniques
- Typically conducted as a "standup", early in the day
 - Status: "What did I do since the last meeting?"
 - Impediments: "Any issues/blockers?"
 - Promises: "What do I promise to resolve today?"



Summary: Scrum is an agile framework

- Should be adapted to fit to the project / work culture / team
- A **Scrum Team** works towards creating product increments
- Adapts well to **change**



Optional Readings

- Ken Schwaber & Jeff Sutherland, The Scrum Guide, 2020
- Takeuchi & Nonaka, *The New New Product Development Game*, *Harvard Business Review*, 1986
- Scrum at scale, <https://www.scrumatscale.com/>
- Scrum product backlog
 - <https://www.scrum.org/resources/what-is-a-product-backlog>

References

- Bruegge, & Dutoit. Object-oriented software engineering. using UML, patterns, and Java. Pearson, 2009.
- Object Management Group. Unified Modeling Language. Version 2.5.1, 2017



Fall 2025

L03 Scrum

CS 1530 Software Engineering

Nadine von Frankenberg



University of
Pittsburgh

School of Computing
and Information

University of Pittsburgh
School of Computing and Information
sci.pitt.edu