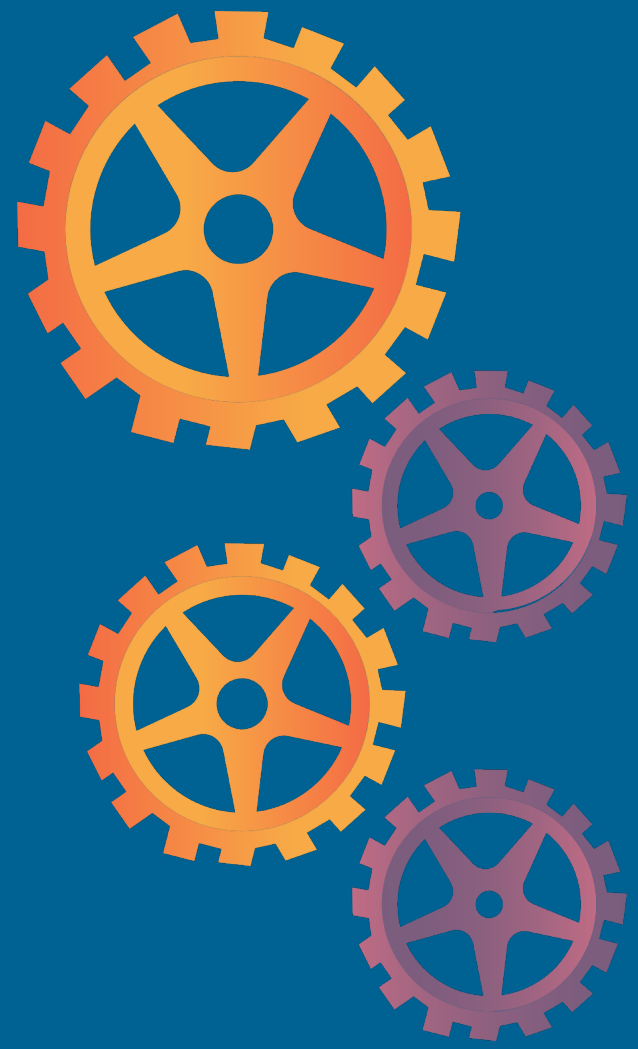
Innovative
Agile Project
Based Learning

Agile Fundamentals

Prof. Panos Fitsilis (fitsilis@uth.gr)
University of Thessaly

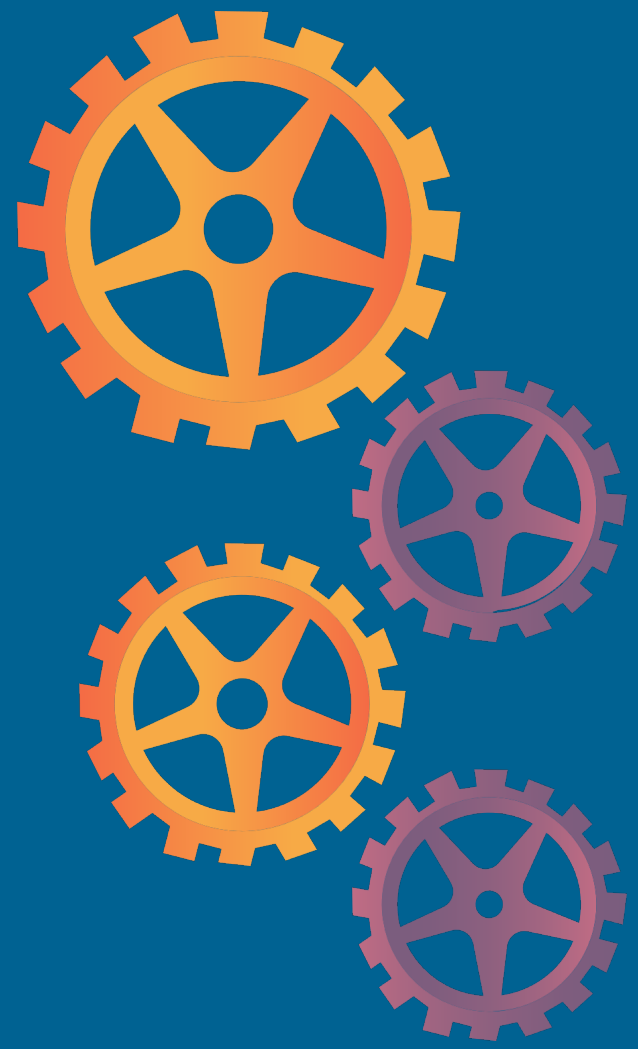
Contents

- Manifesto for Agile Methods
- Principles behind Agile Manifesto
- Empiricism is the foundation of Scrum
- Scrum at a glance



The Agile Manifesto

Individuals and interactions	instead	Processes and tools
Working software	instead	Comprehensive documentation
Customer collaboration	instead	Contract negotiation
Responding to change	instead	Following a plan



Principles over agile manifesto- (1)

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

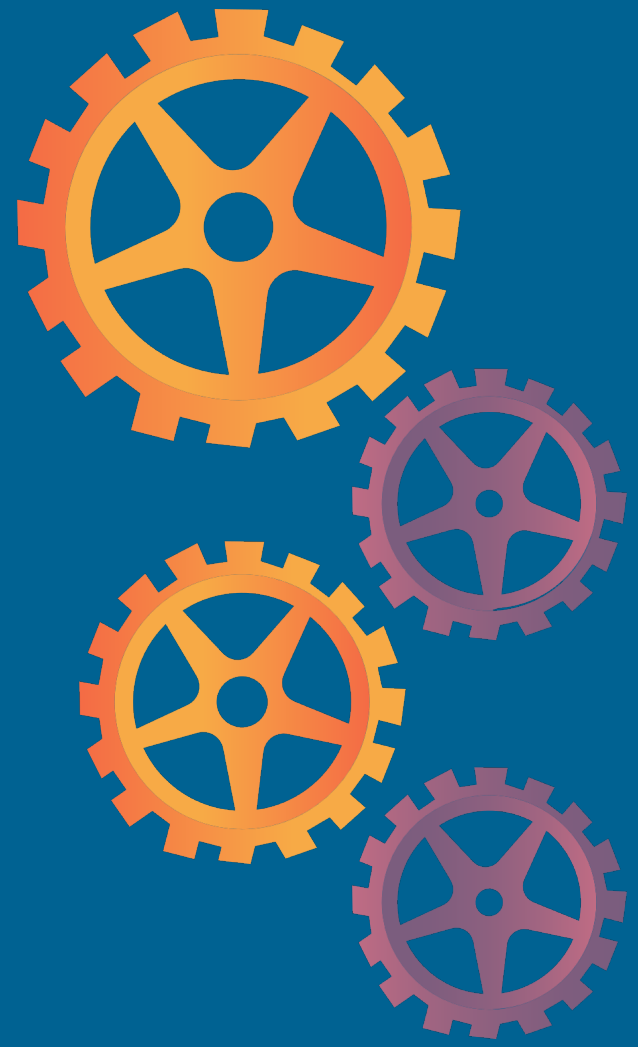
2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

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.



Principles over agile manifesto (2)

7 Working software is the primary measure of progress.

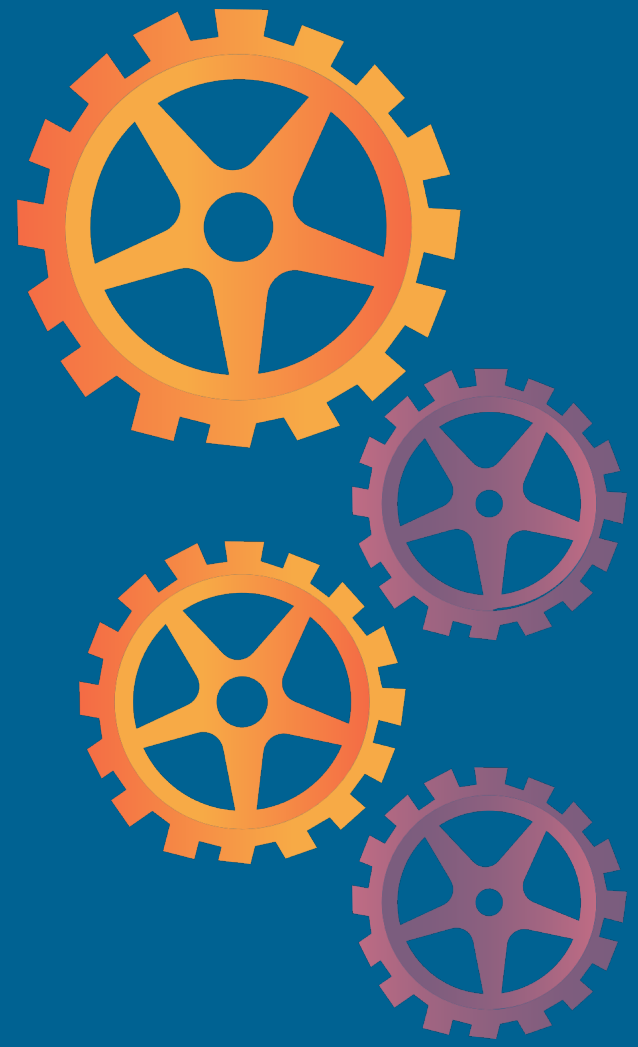
8 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9 Continuous attention to technical excellence and good design enhances agility.

10 Simplicity–the art of maximizing the amount of work not done–is essential.

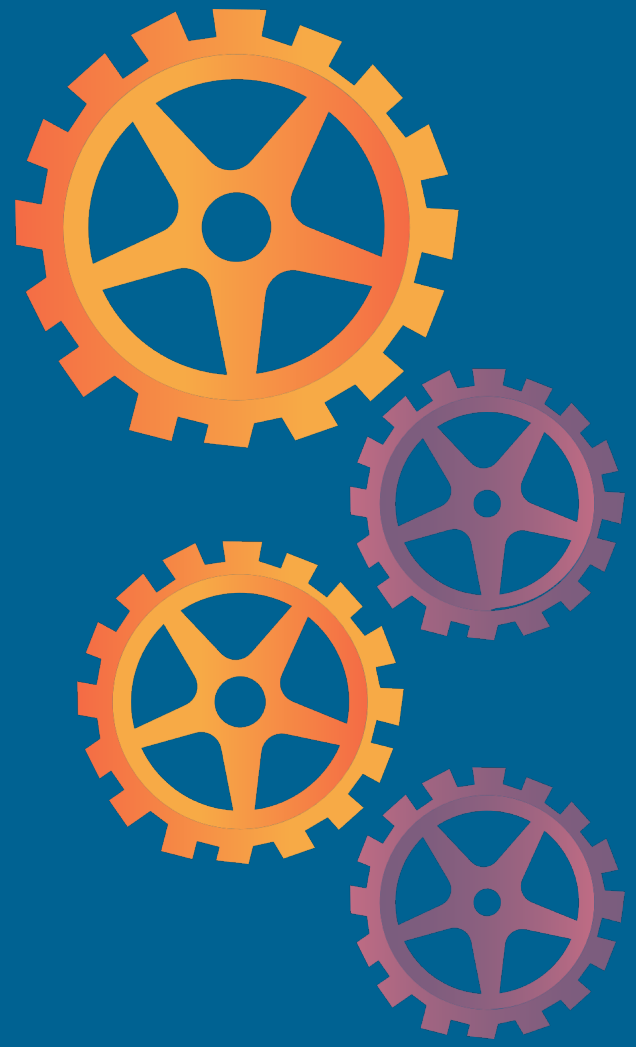
11 The best architectures, requirements, and designs emerge from self-organizing teams.

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

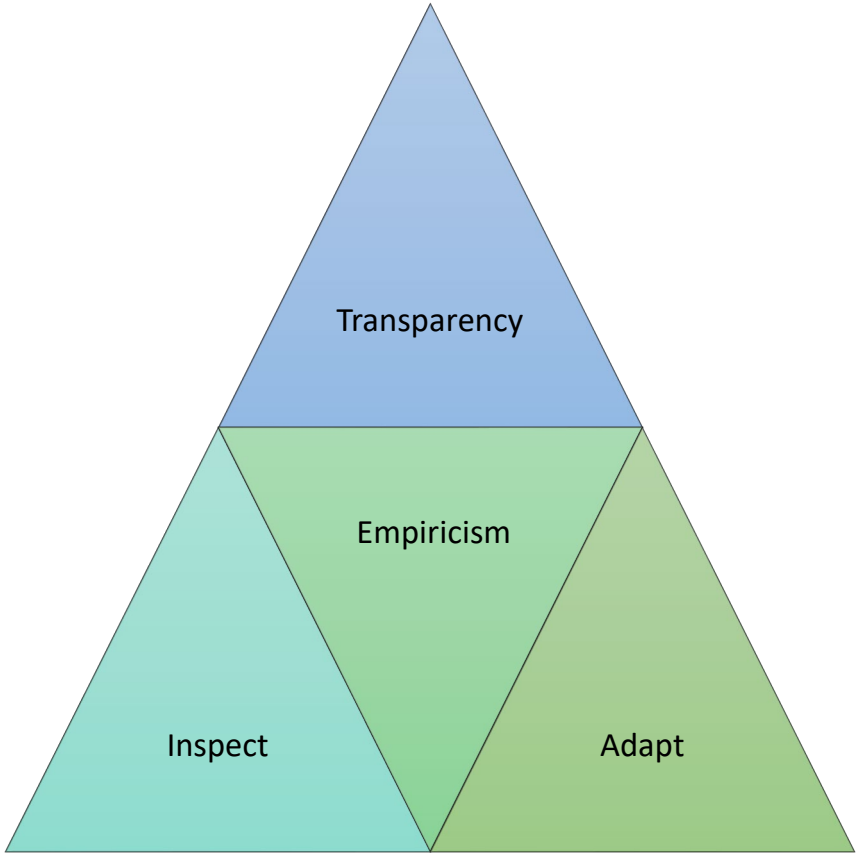


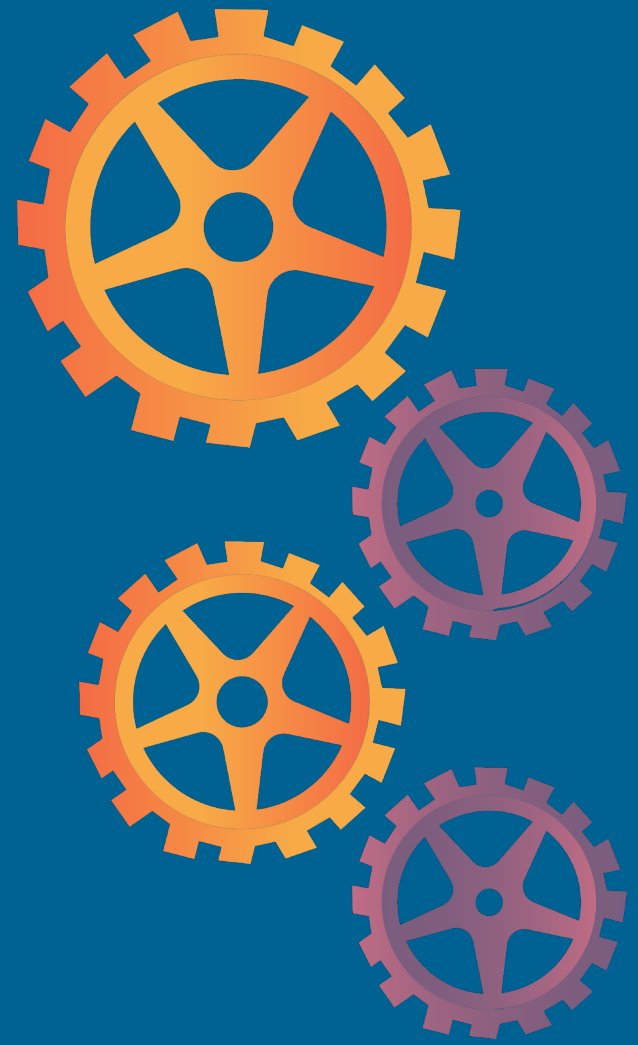
Question for discussion #1

- Which of the agile principles find more relevant with your work and why?
- How this principle will be applied in a school?



Empiricism is the foundation of agile methods





Innovative
Agile Project
Based Learning

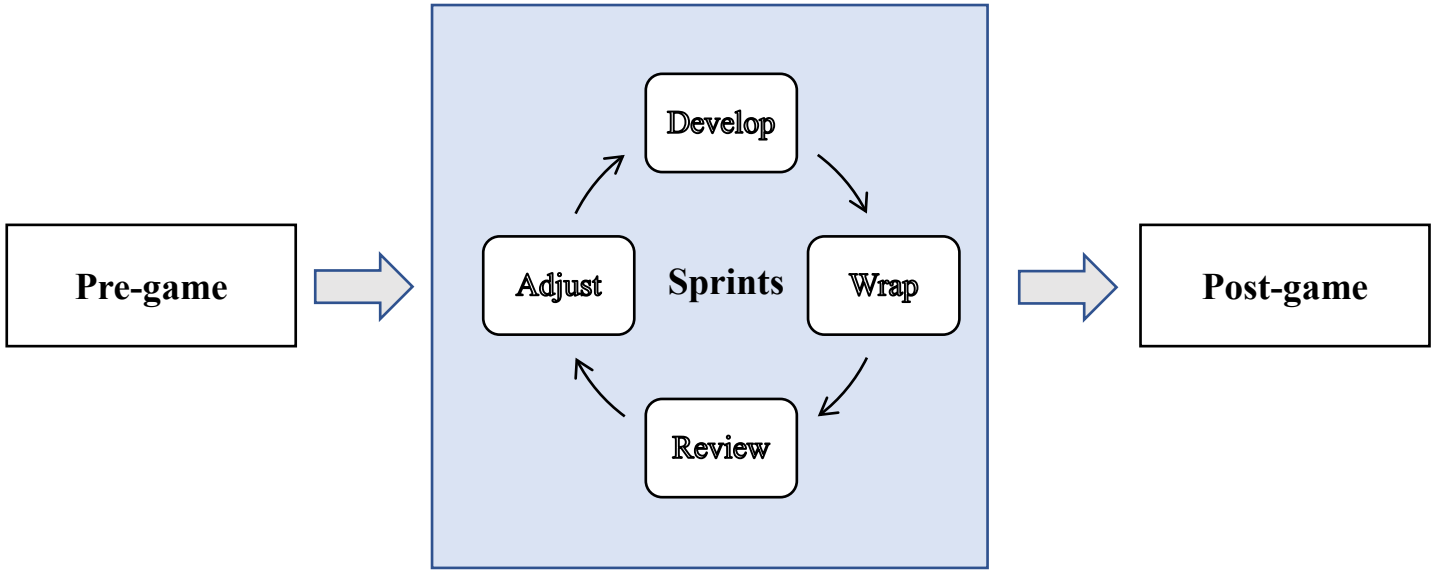
Scrum

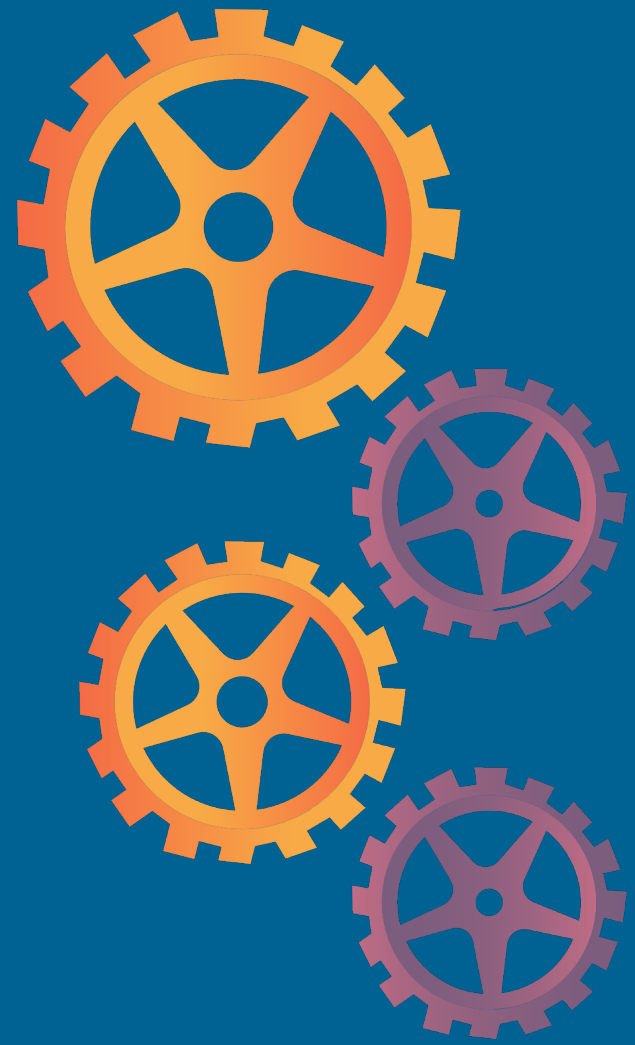
Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- Embedded systems

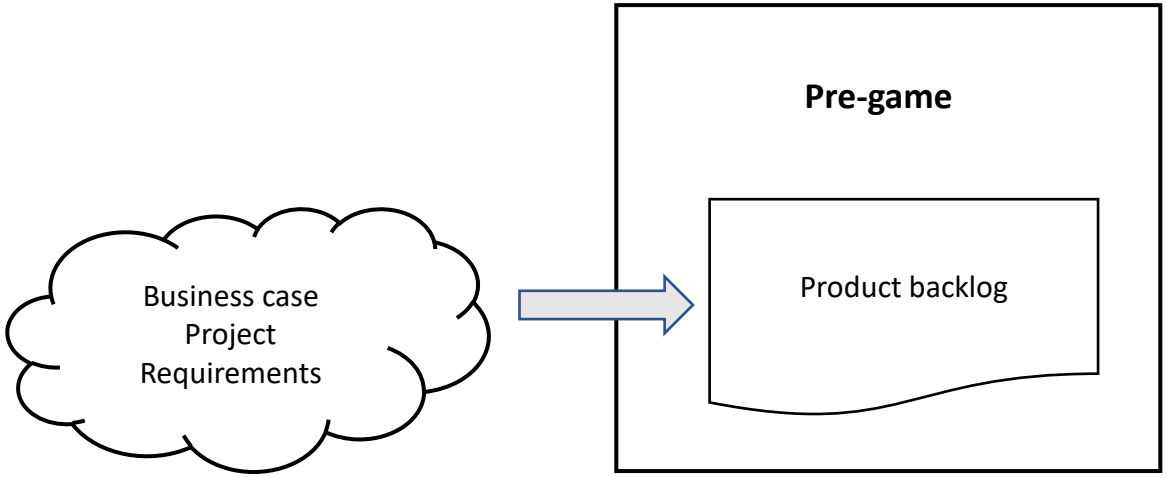
- Marketing projects
- Financial projects
- Product development project
- Educational projects

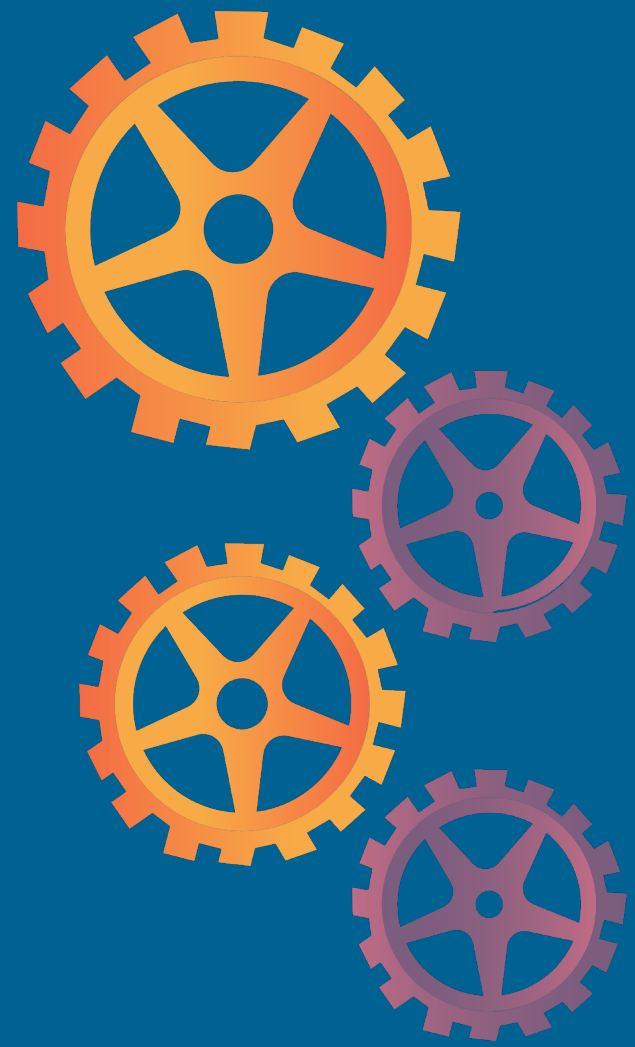
Scrum phases



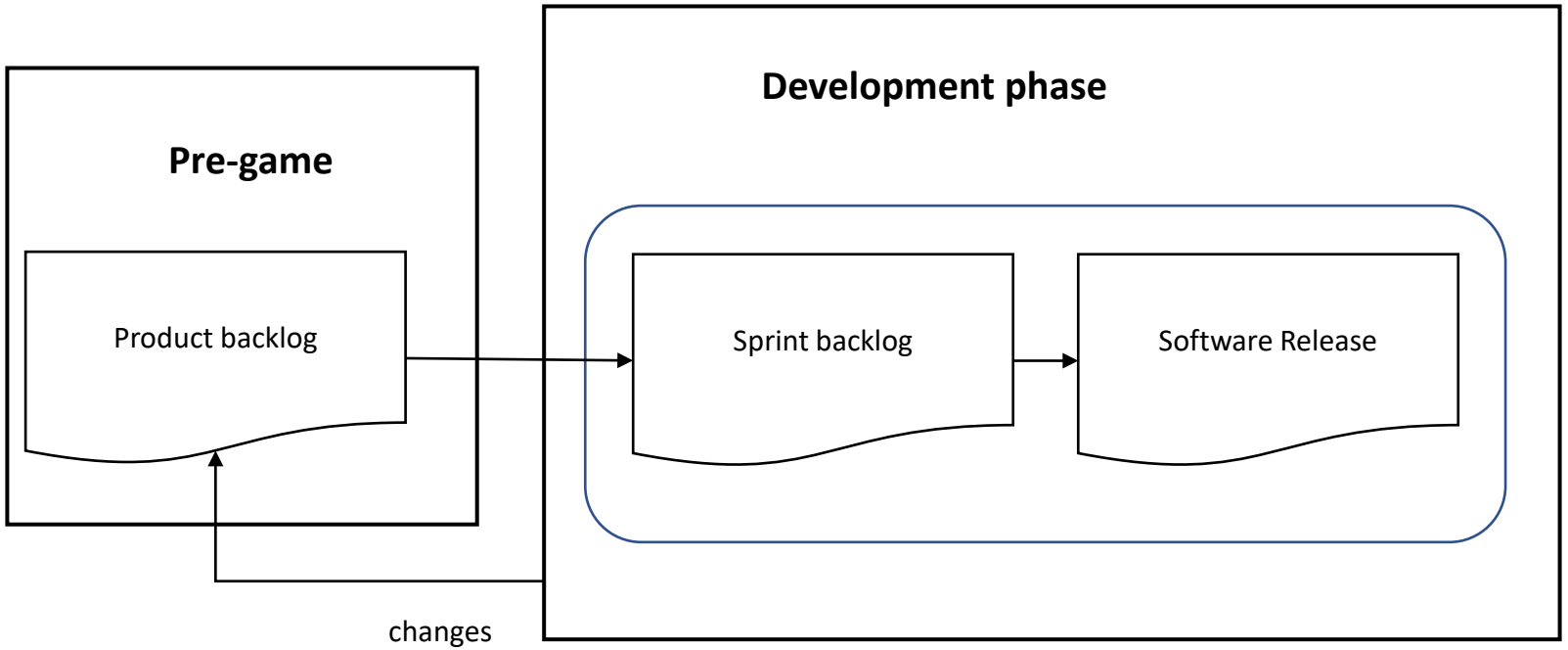


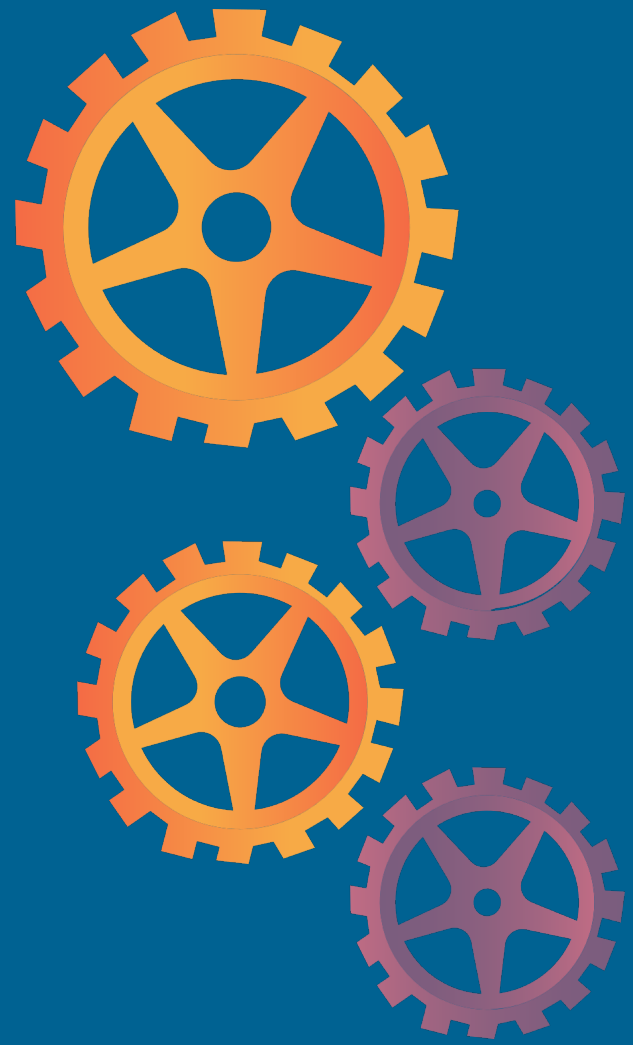
Pregame phase – Project Inception



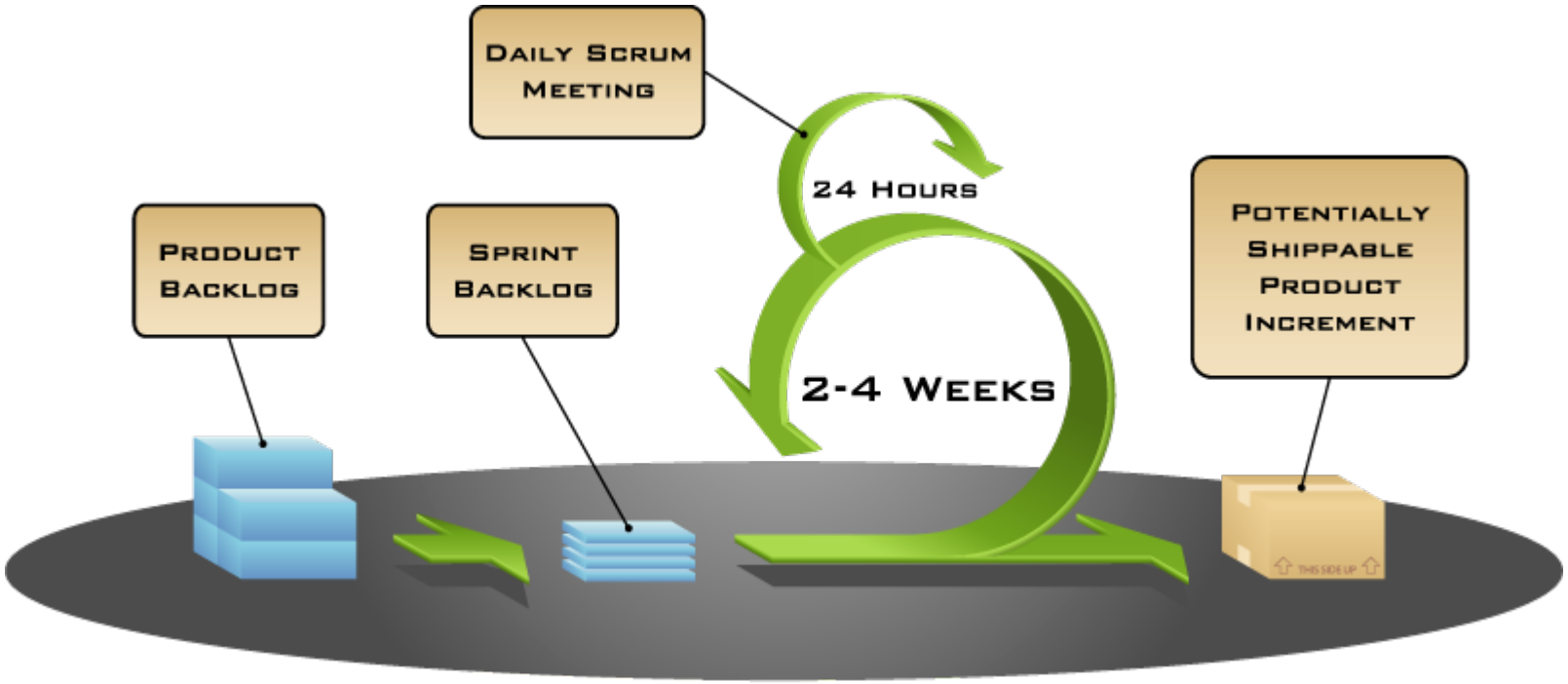


Development phase





Scrum at a glance



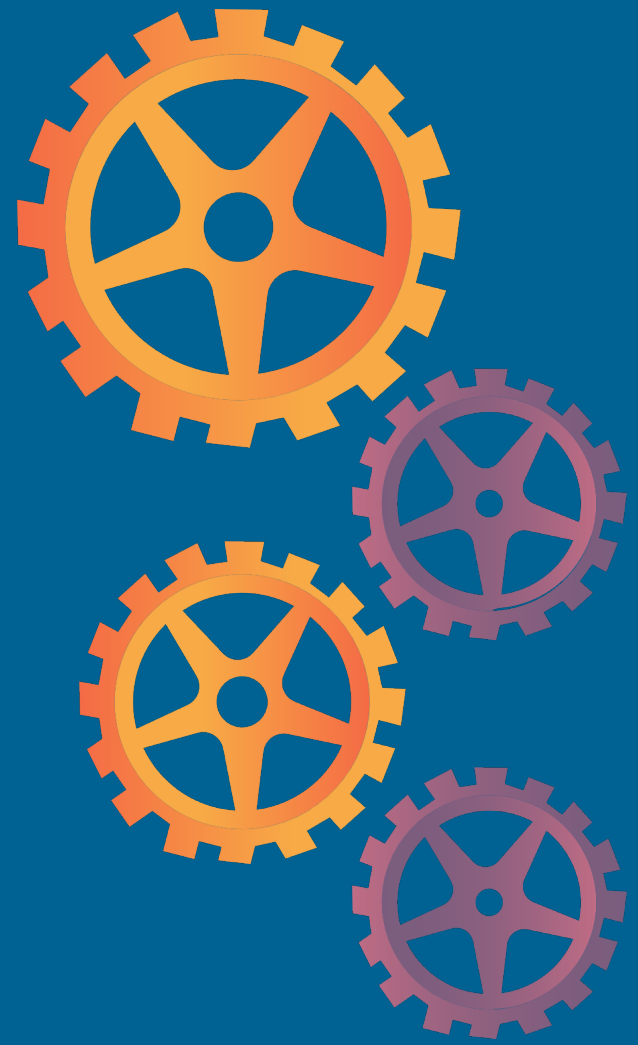
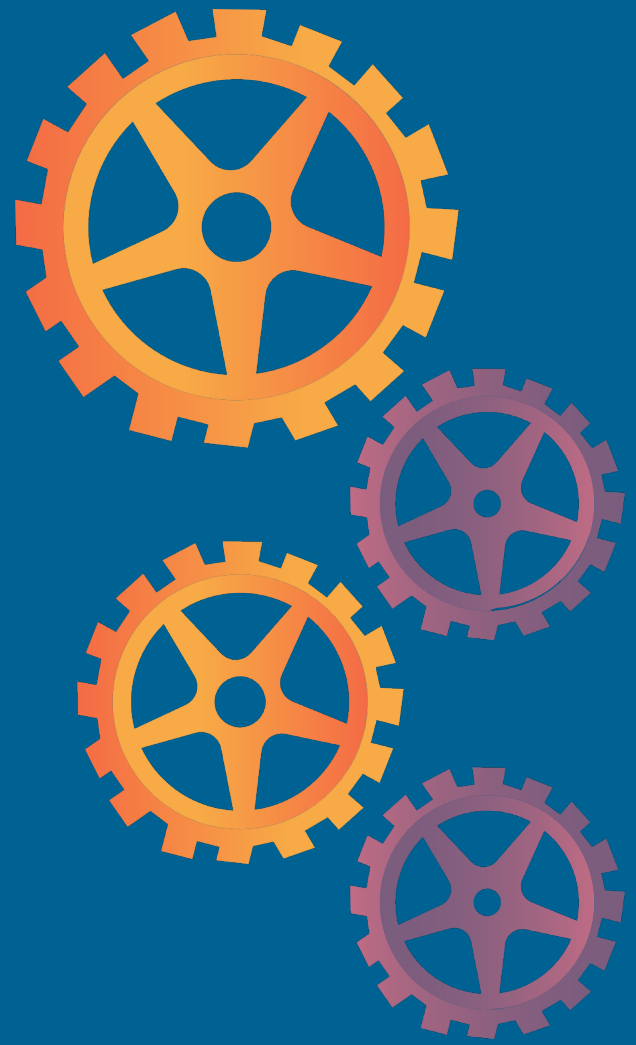


Image available at www.mountaingoatsoftware.com/scrum

Sprints

- Scrum projects make progress in a series of “sprints”
  - Analogous to Extreme Programming iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

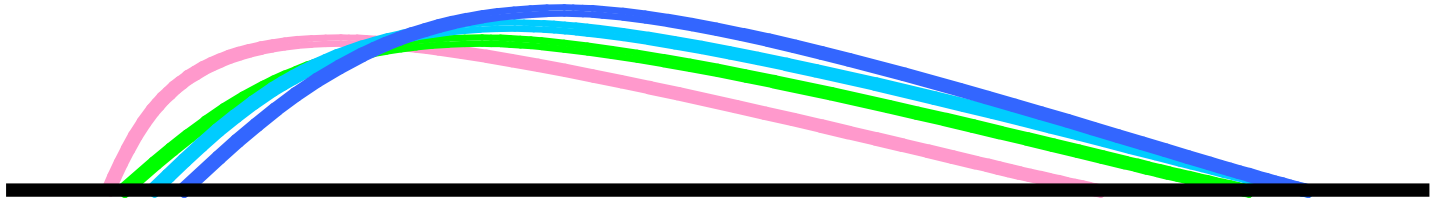


Sequential vs. overlapping development

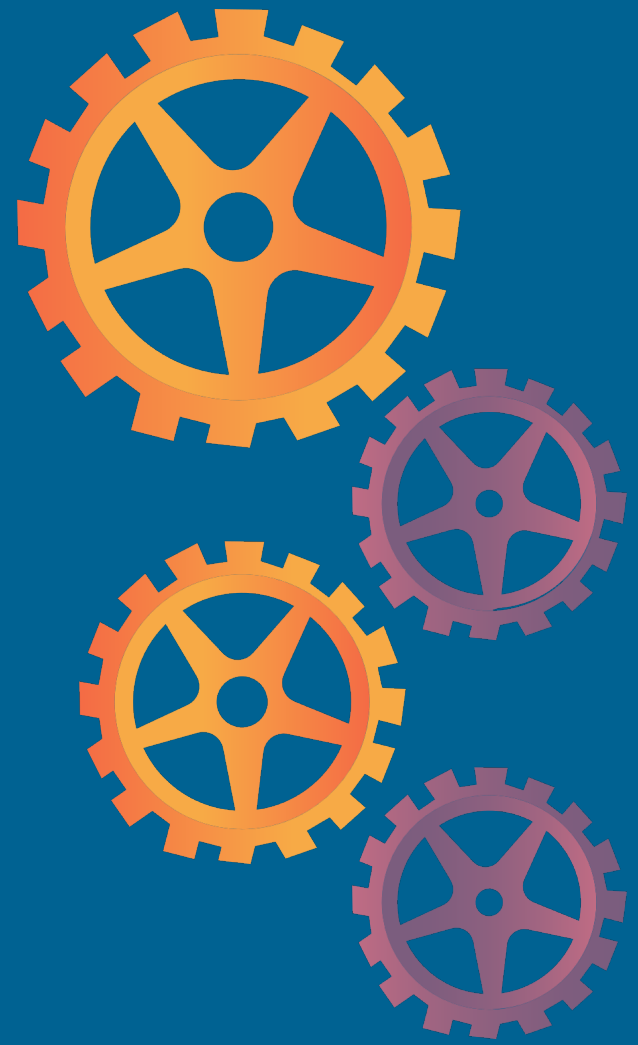
Requirements | Design | Implement | Test

Rather than doing all of one thing at a time...

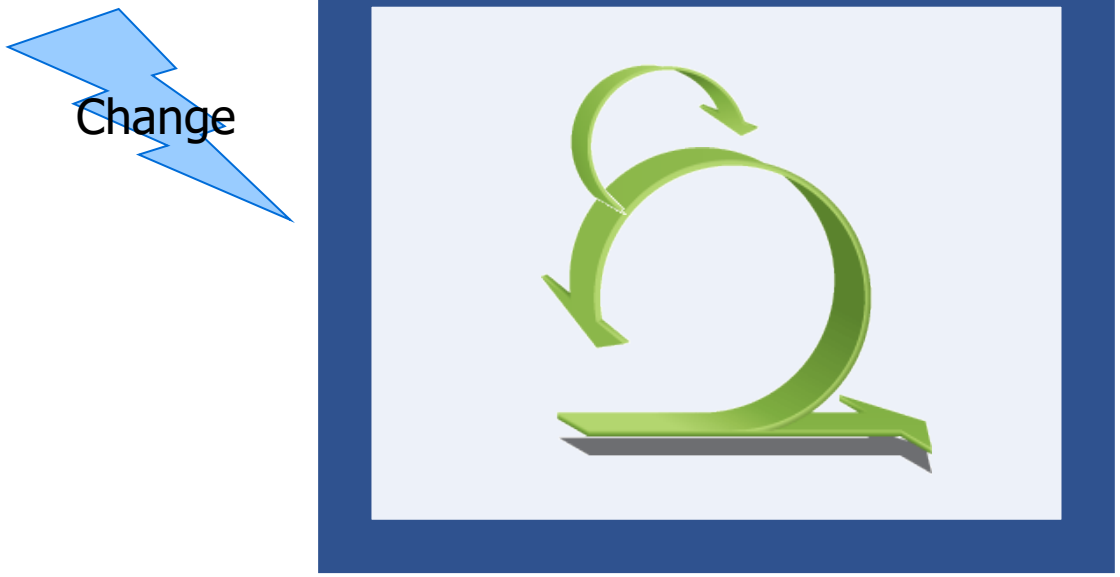
...Scrum teams do a little of everything all the time



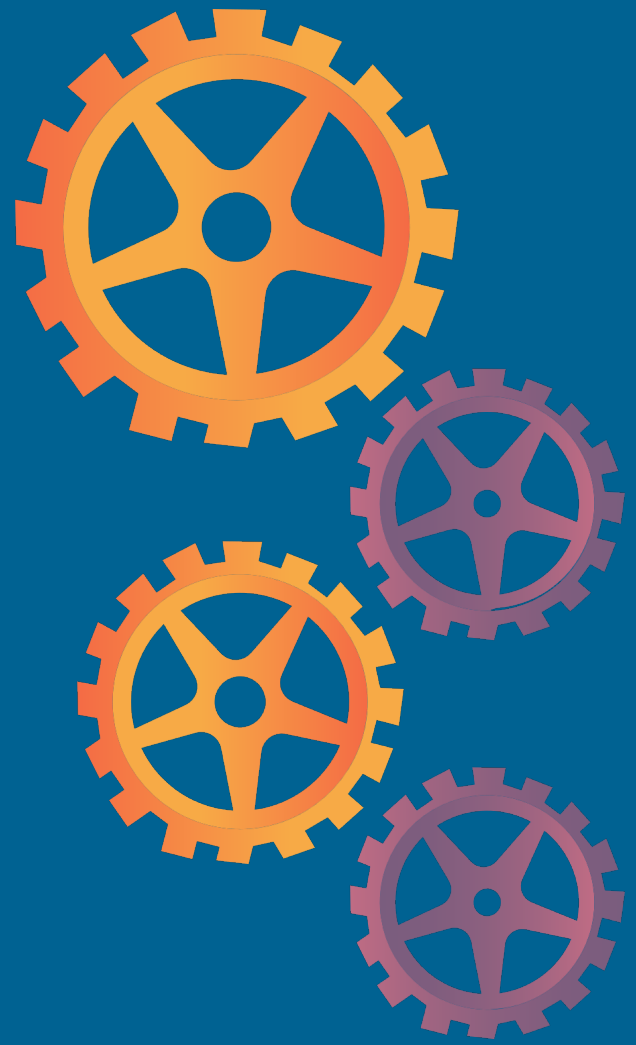
Source: "The New envelopment Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review,* January 1986.



No changes during a sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint



Scrum framework

Roles

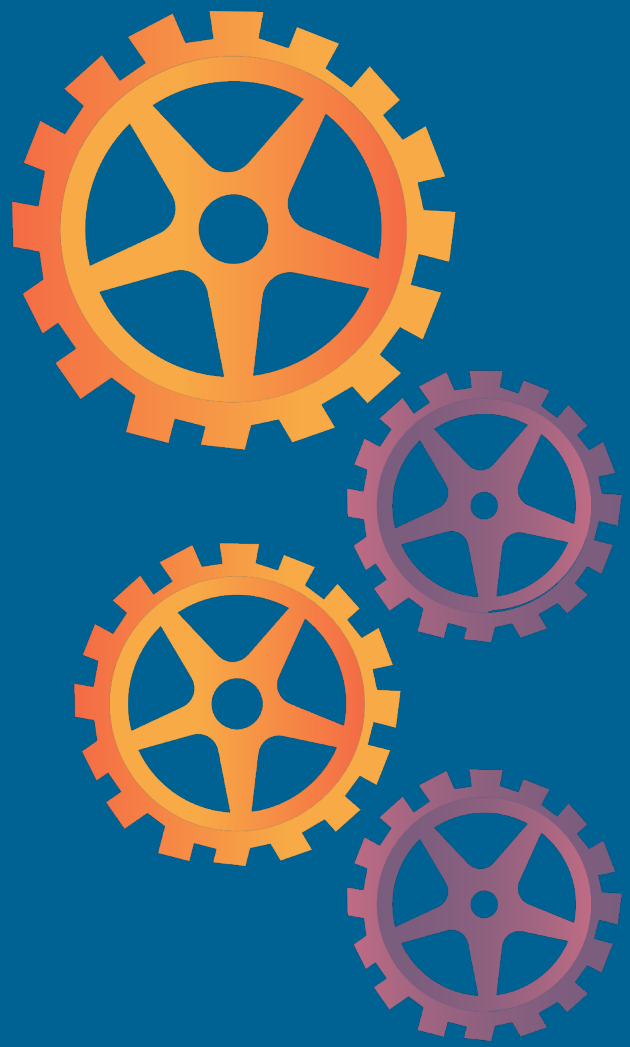
- Product owner
- ScrumMaster
- Team

Ceremonies

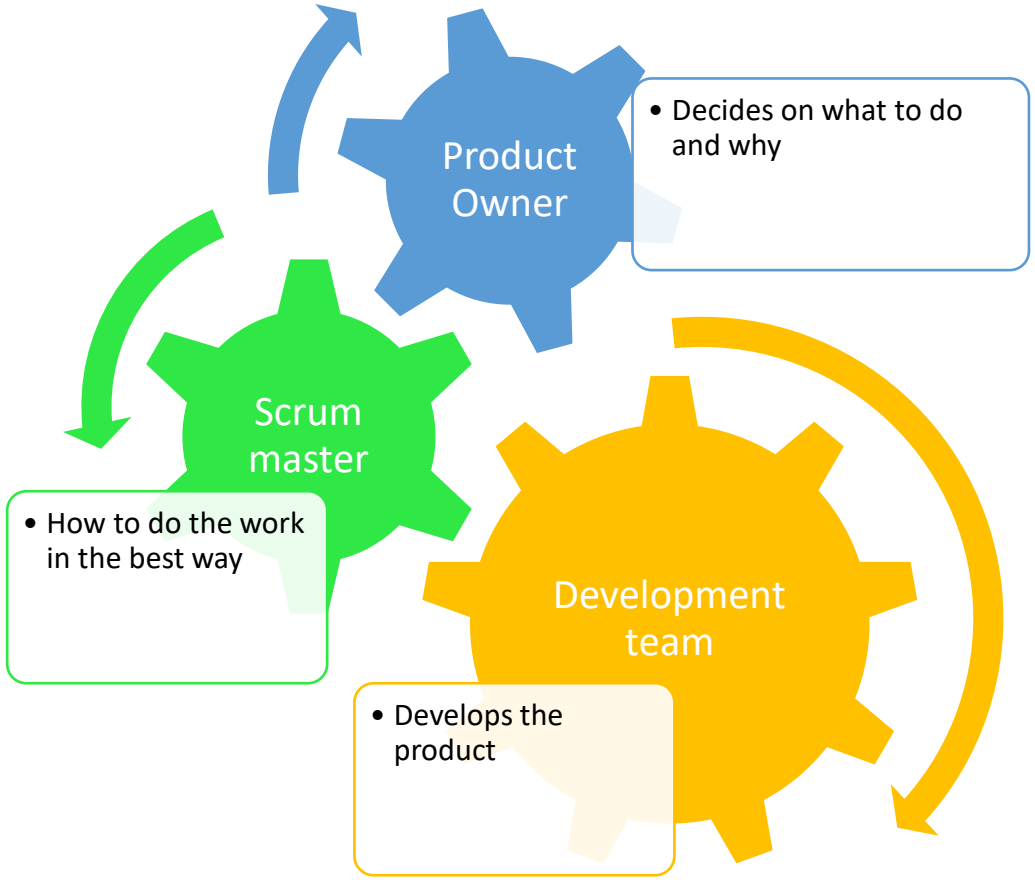
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

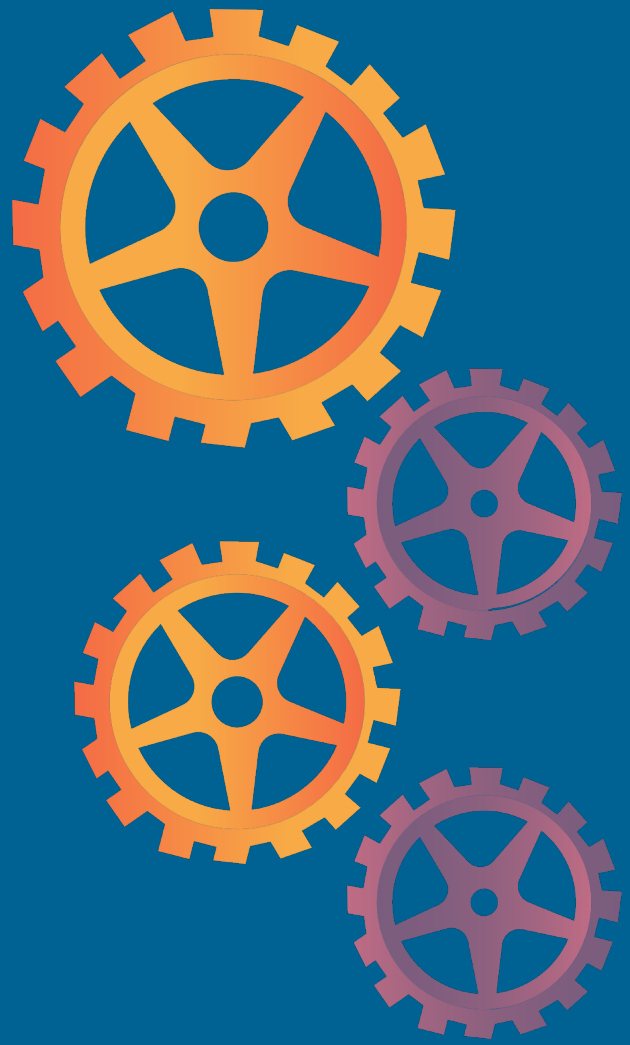
Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



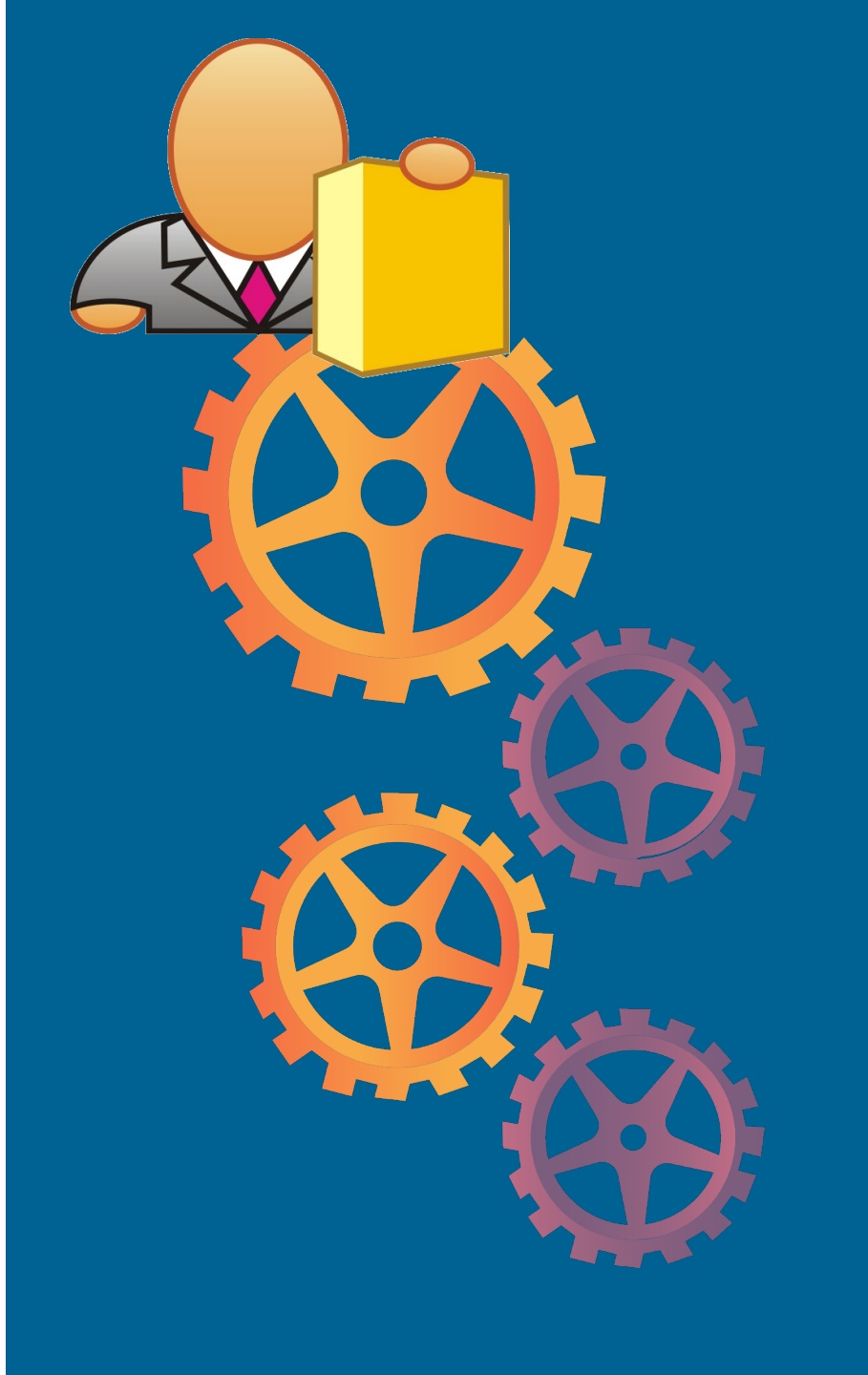
Main roles in Scrum





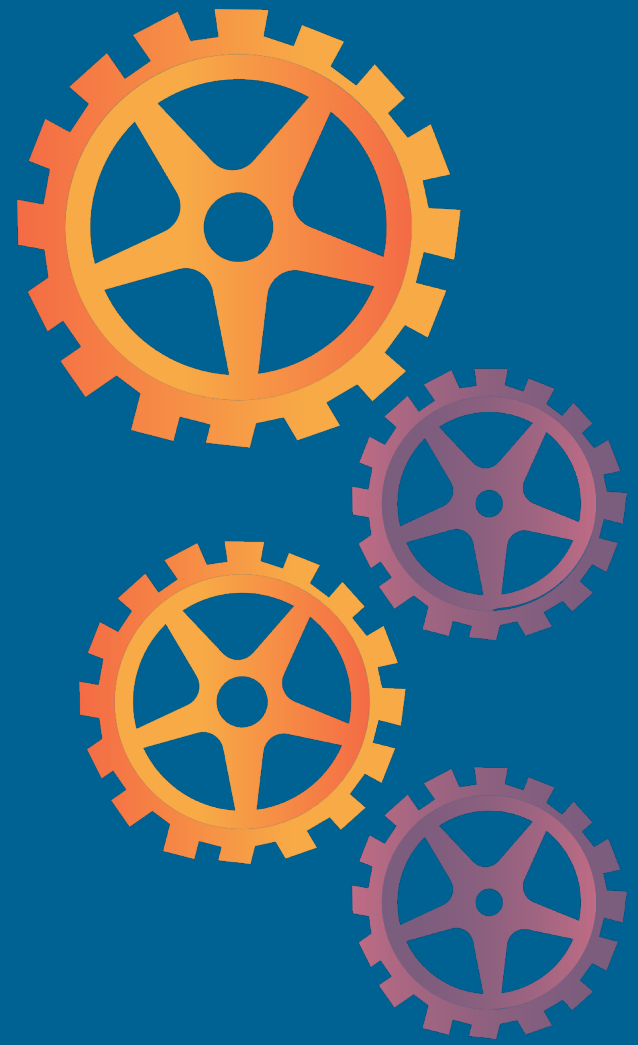
Product owner

- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results



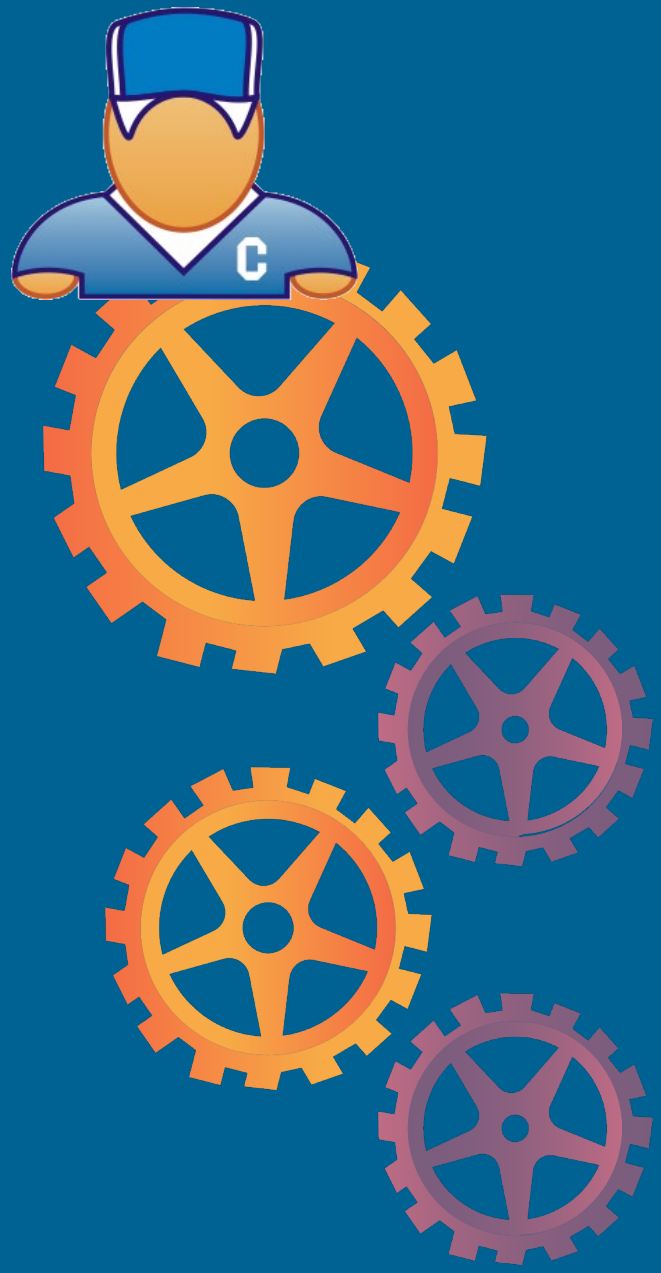
Product owner tasks

- Creates product vision
- Manages Product Backlog
- Prioritizes requirements
- Supervises work
- Proposes changes
- Is the liaison between customer and team
- Evaluates the work at the end of the sprint
- Manages project financials

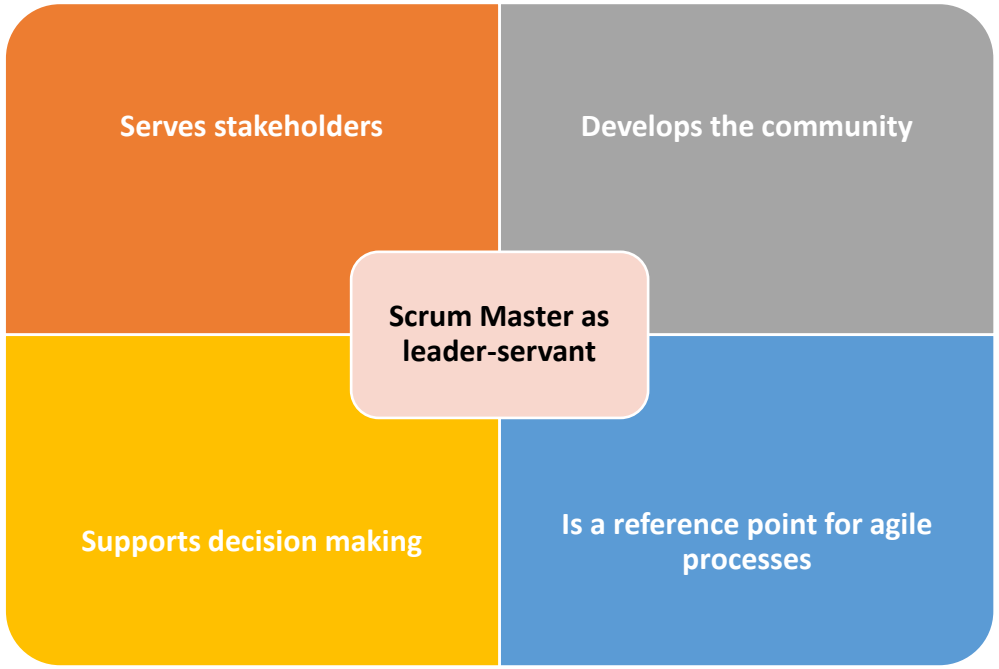


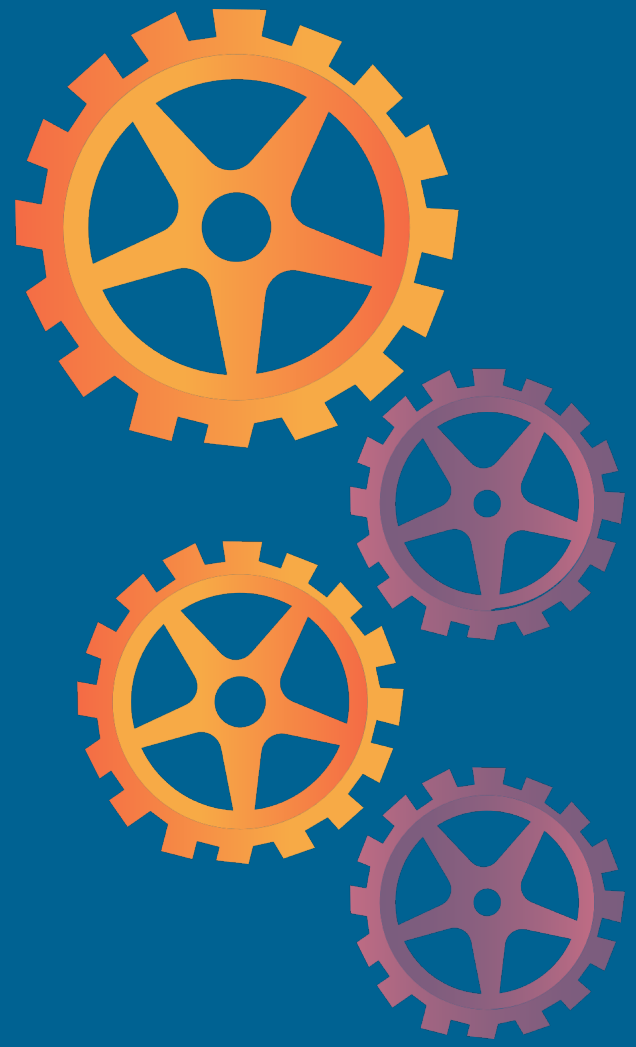
The ScrumMaster

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences



Scrum master as leader-servant

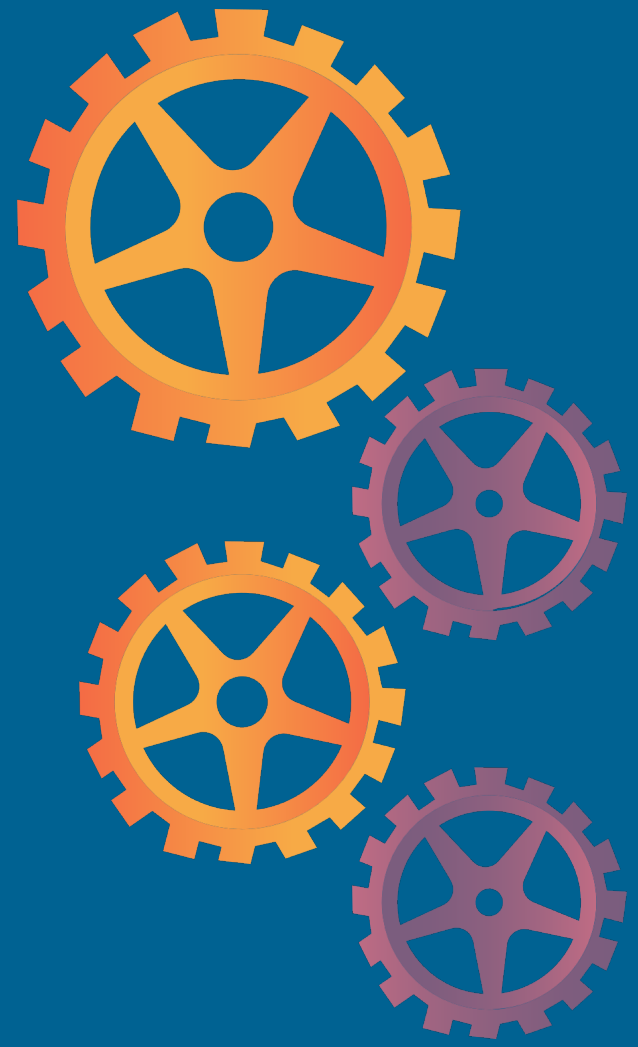




Question for discussion #2

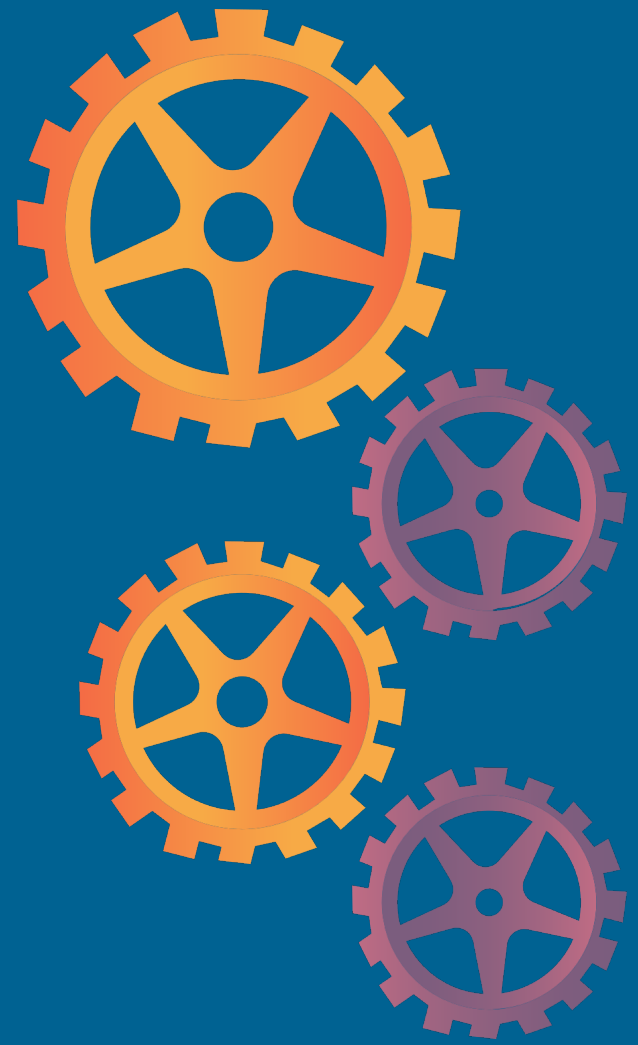
- According to your knowledge what are the other leadership styles you are aware of?

- What leadership style is most appropriate for a school project



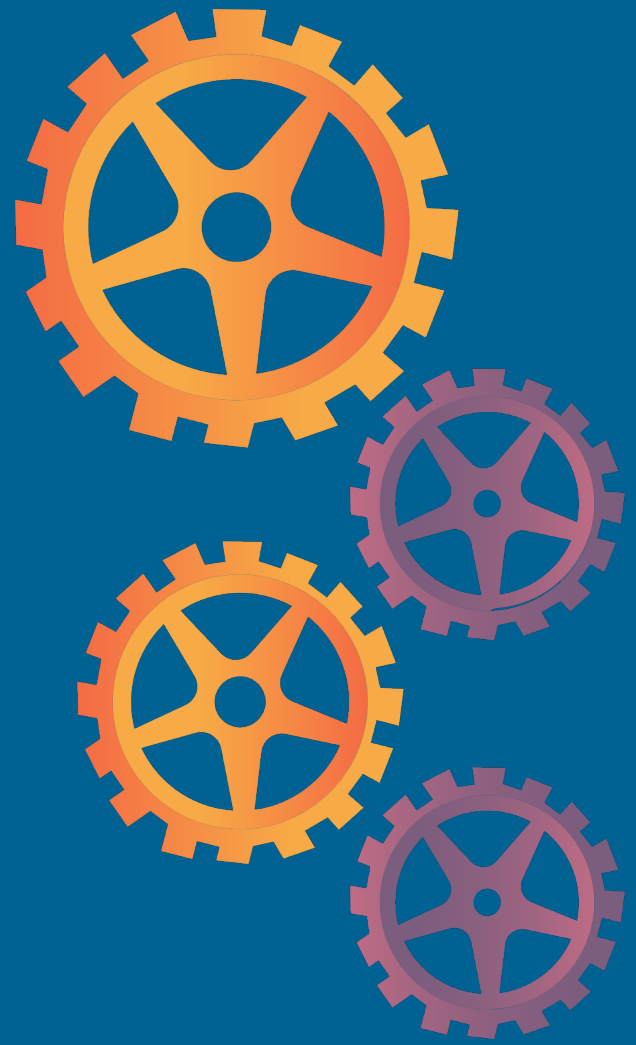
The team

- Typically 5-9 people
- Cross-functional:
  - Programmers, testers, user experience designers, etc.
- Members should be full-time
  - May be exceptions (e.g., database administrator)



The team

- Teams are self-organizing
  - Ideally, no titles but rarely a possibility
- Membership should change only between sprints



Scrum framework

Roles

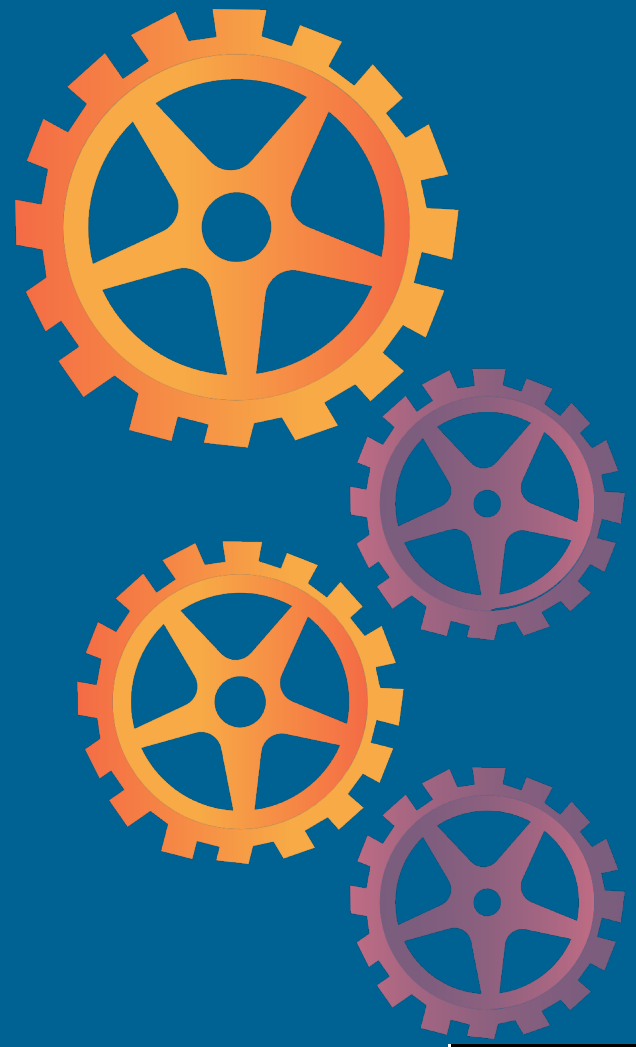
- Product owner
- ScrumMaster
- Team

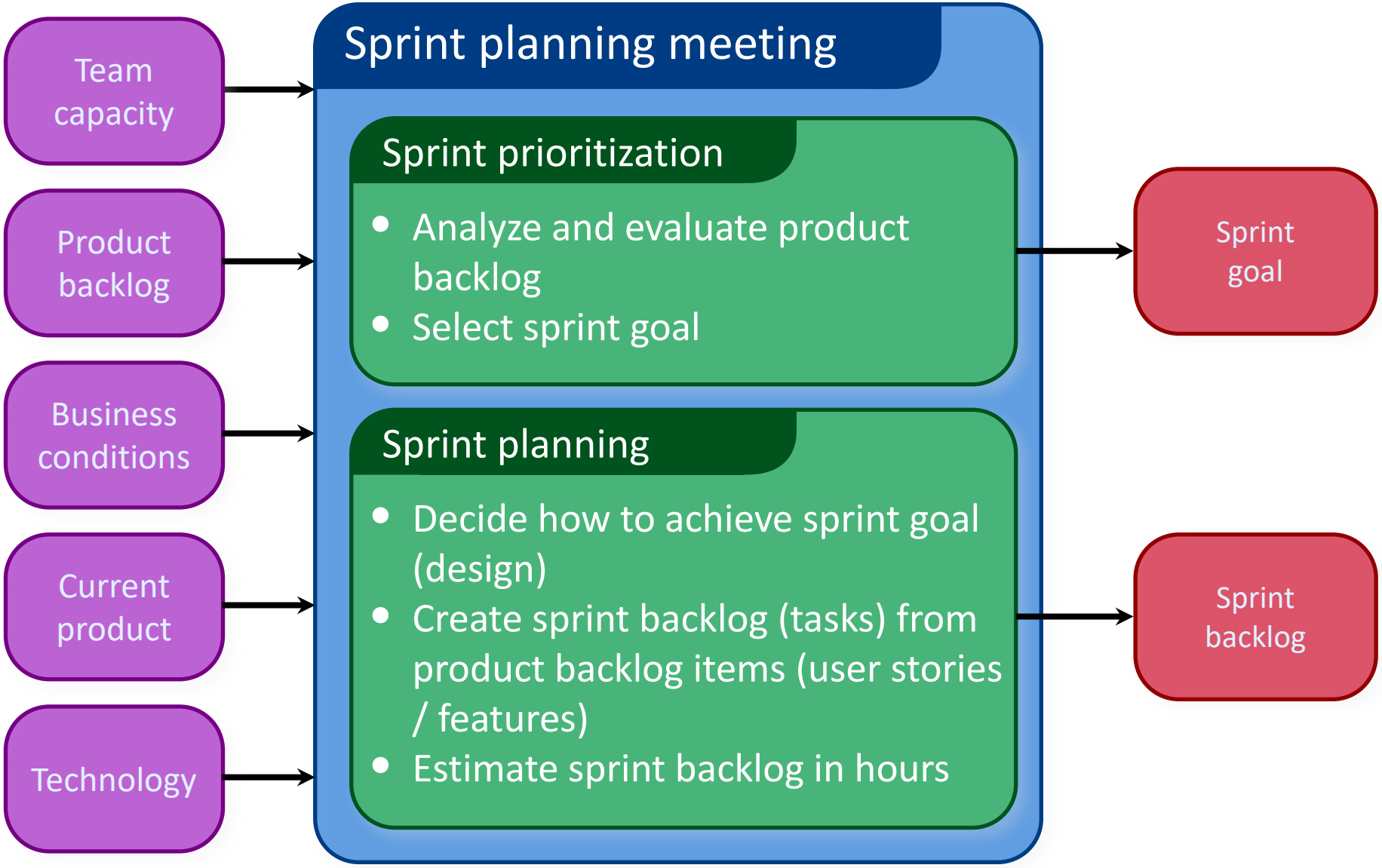
Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

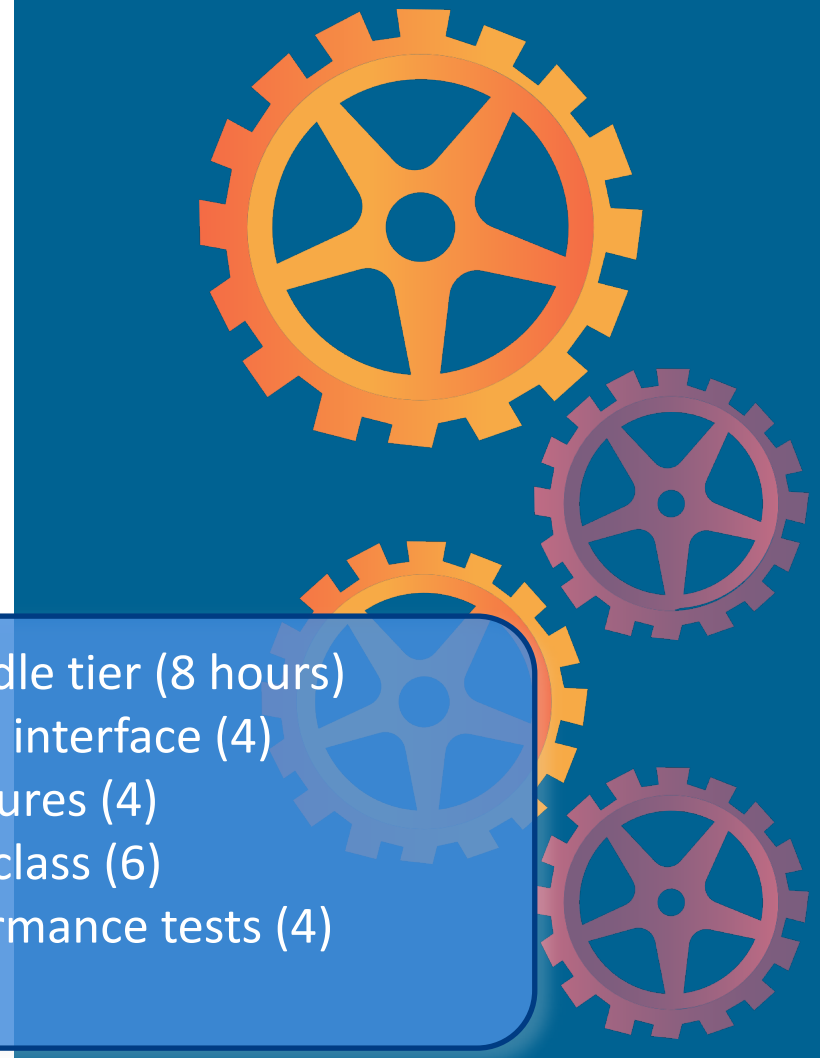
- Product backlog
- Sprint backlog
- Burndown charts




Team capacity
Product backlog
Business conditions
Current product
Technology
Sprint planning meeting
Sprint prioritization
Analyze and evaluate product backlog
Select sprint goal
Sprint planning
Decide how to achieve sprint goal (design)
Create sprint backlog (tasks) from product backlog items (user stories / features)
Estimate sprint backlog in hours
Sprint goal
Sprint backlog

Sprint planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
  - Tasks are identified and each is estimated (1-16 hours)
  - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered



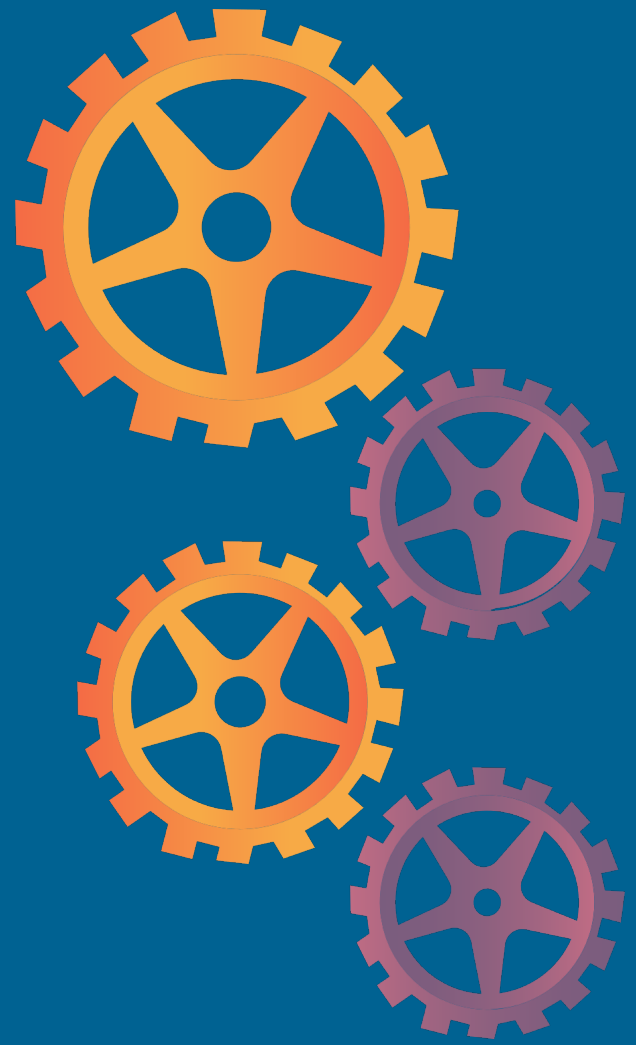
As a vacation planner, I want to see photos of the hotels.

→

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

The daily scrum

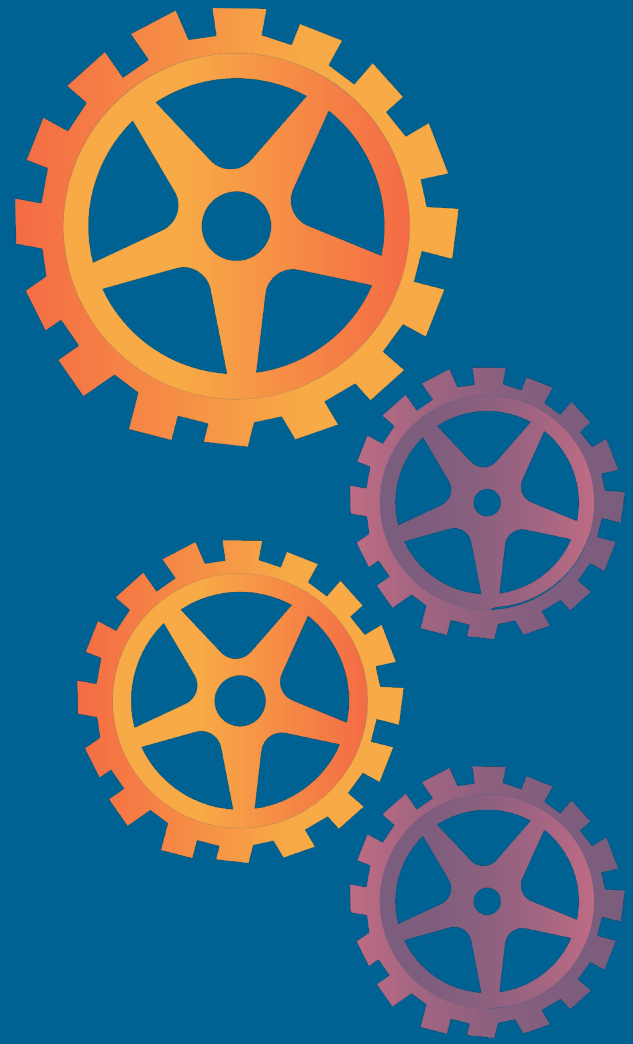
- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings



Everyone answers 3 questions

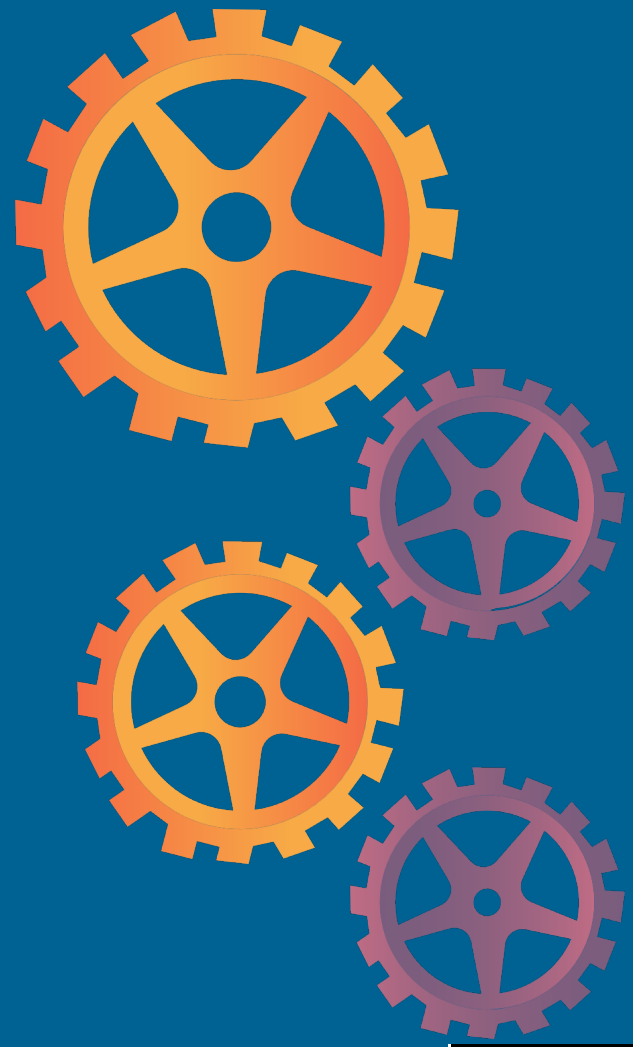
- These are *not* status for the ScrumMaster
  - They are commitments in front of peers

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?



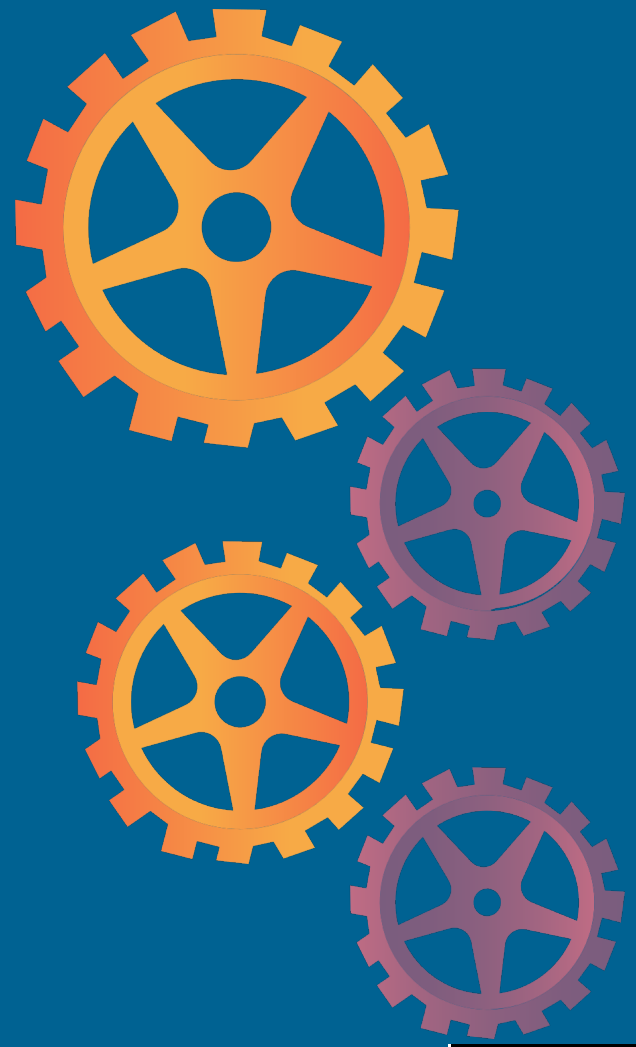
The sprint review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Invite the world



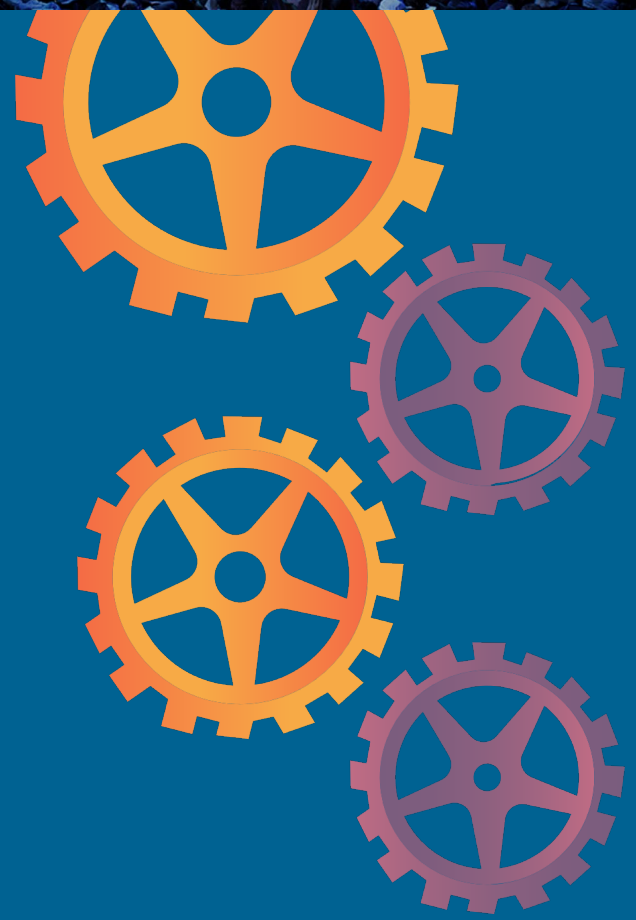
Sprint retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - ScrumMaster
  - Product owner
  - Team
  - Possibly customers and others



Scrum: How to do twice as much in half the time

https://www.youtube.com/watch?v=s4thQcgLCqk

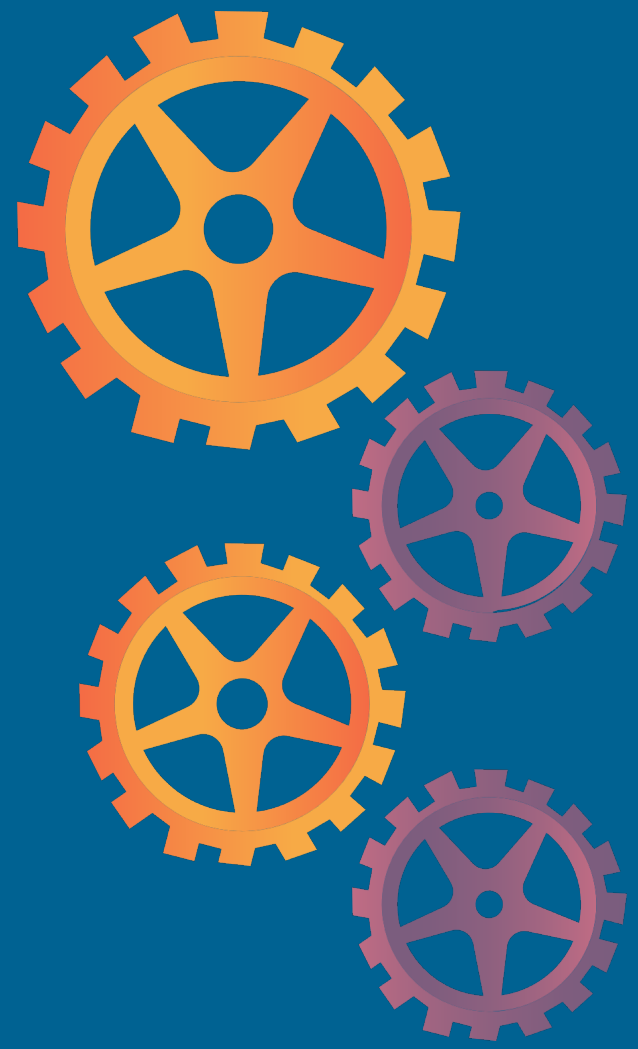


Innovative
Agile Project
Based Learning

Other agile techniques

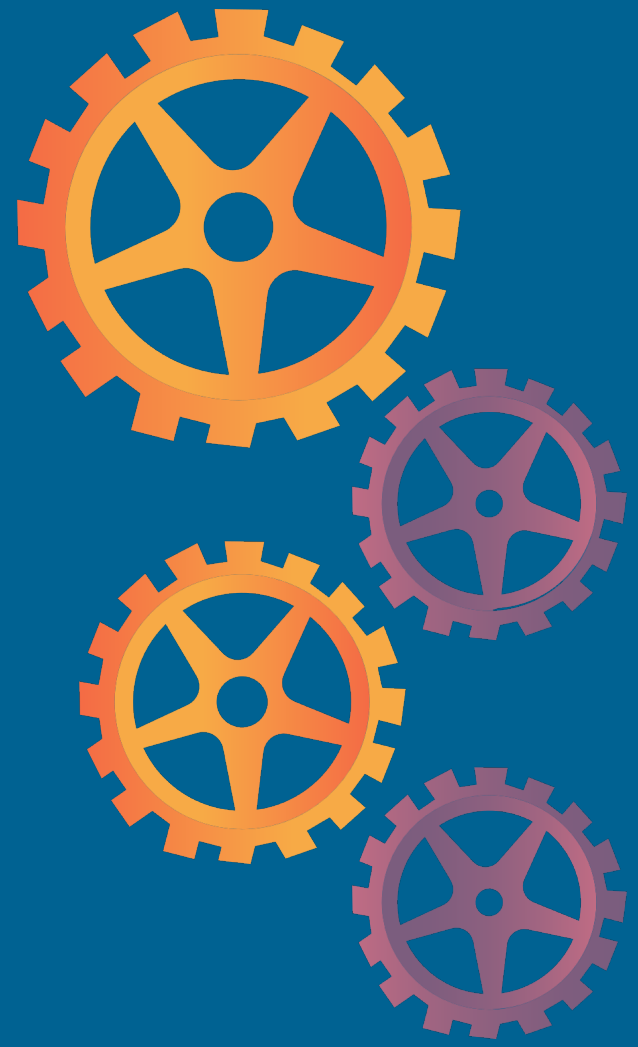
Lean development

- **Lean Software Development** is an agile framework based on optimizing development time and resources, eliminating waste, and ultimately delivering only what the product needs.
  - The Lean approach is also often referred to as the **Minimum Viable Product (MVP)** strategy, in which a team releases a bare-minimum version of its product to the market, learns from users what they like, don't like and want to be added, and then iterates based on this feedback.
- The main difference between lean and other agile methods is that Lean focuses on optimizing project products through building better processes while agile focuses on building better products directly.
- Another important difference is that
  - Lean is continuous
  - Scrum is iterative



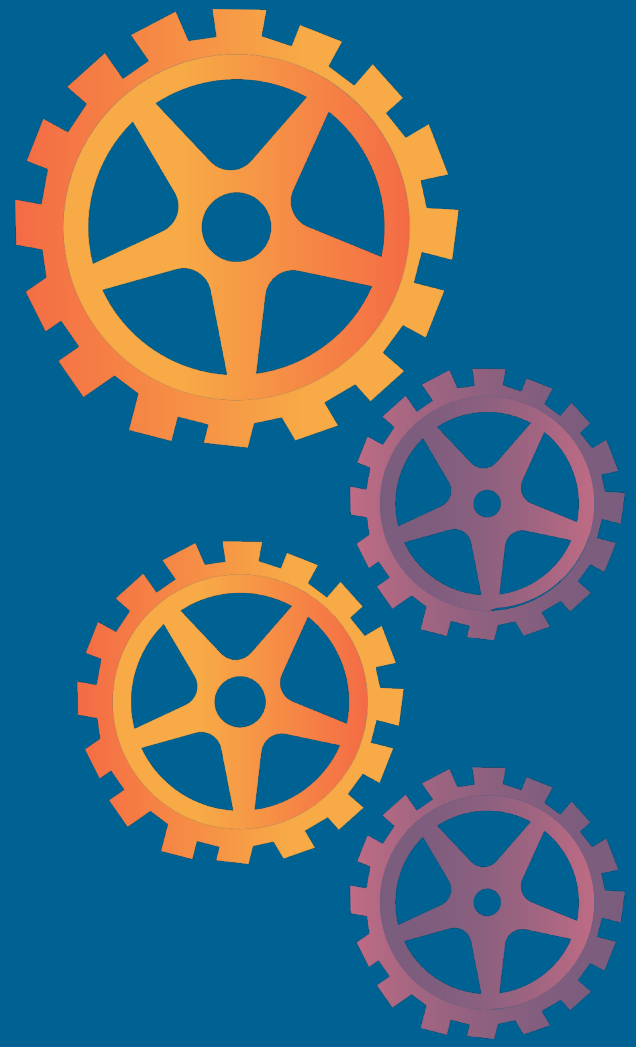
Lean development principles /1

- **Eliminate waste**. Anything that does not add value to the product should be eliminated. E.g.: unnecessary code or software features, bureaucratic processes, quality issues etc.
- **Build in quality**. Various approaches are used to ensure quality is built into the Lean process, such as pair programming and test-driven development.
- **Amplify learning**. Any knowledge gained by a team member should be shared to other members of the development team.
- **Delay commitment as long as possible**. Commitment to irreversible decisions should be delayed as long as possible until a sufficient amount of knowledge is gained on the subject matter. Developers incorporate features and functionality as late as possible in the process to prevent having to redo work as the market changes.



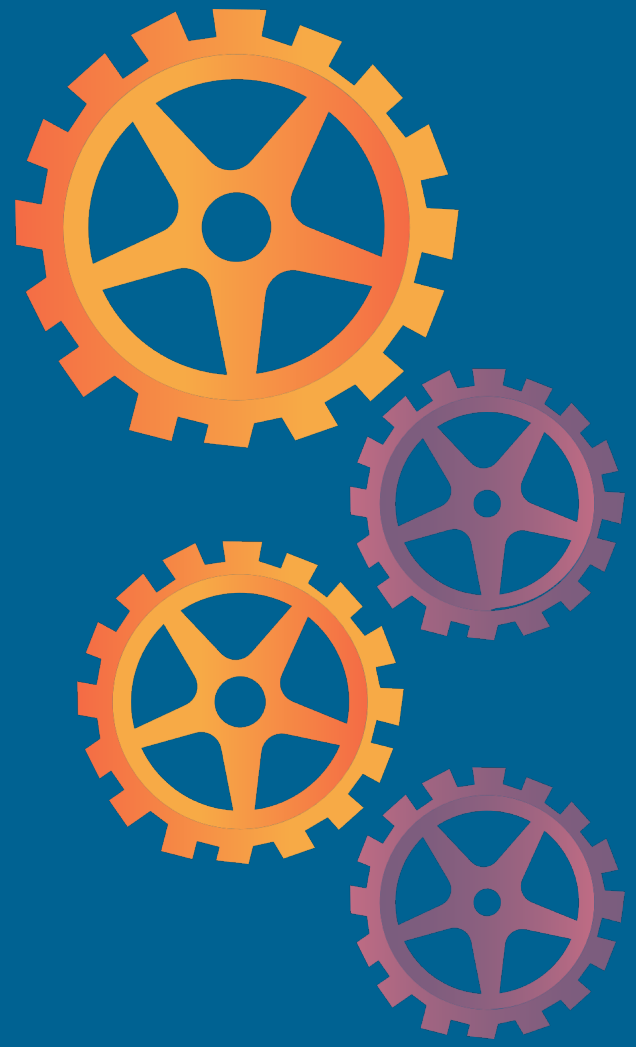
Lean development principles /2

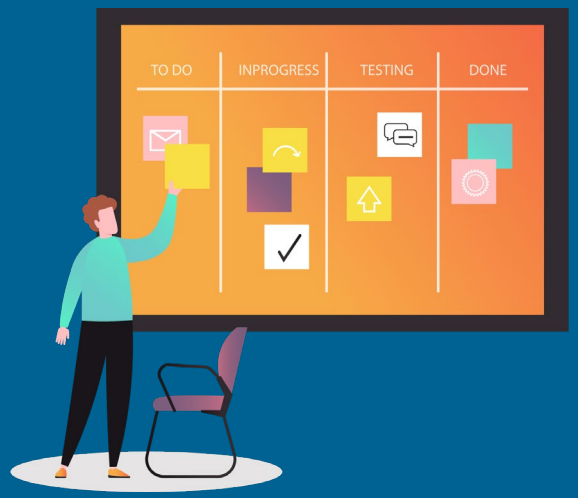
- **Deliver fast.** Developers launch a product quickly them receive customer feedback fast and use that feedback to create a strategy for improvement. The idea behind this that failures will be identified quickly.
- **Respect people.** Respect is the basis for a productive, collaborative atmosphere. Lean encourages healthy conflict, proactive communication and constant feedback.
- **Optimize the whole.** The team examines the entire process to make the Lean process as efficient as possible.



Question for discussion #3

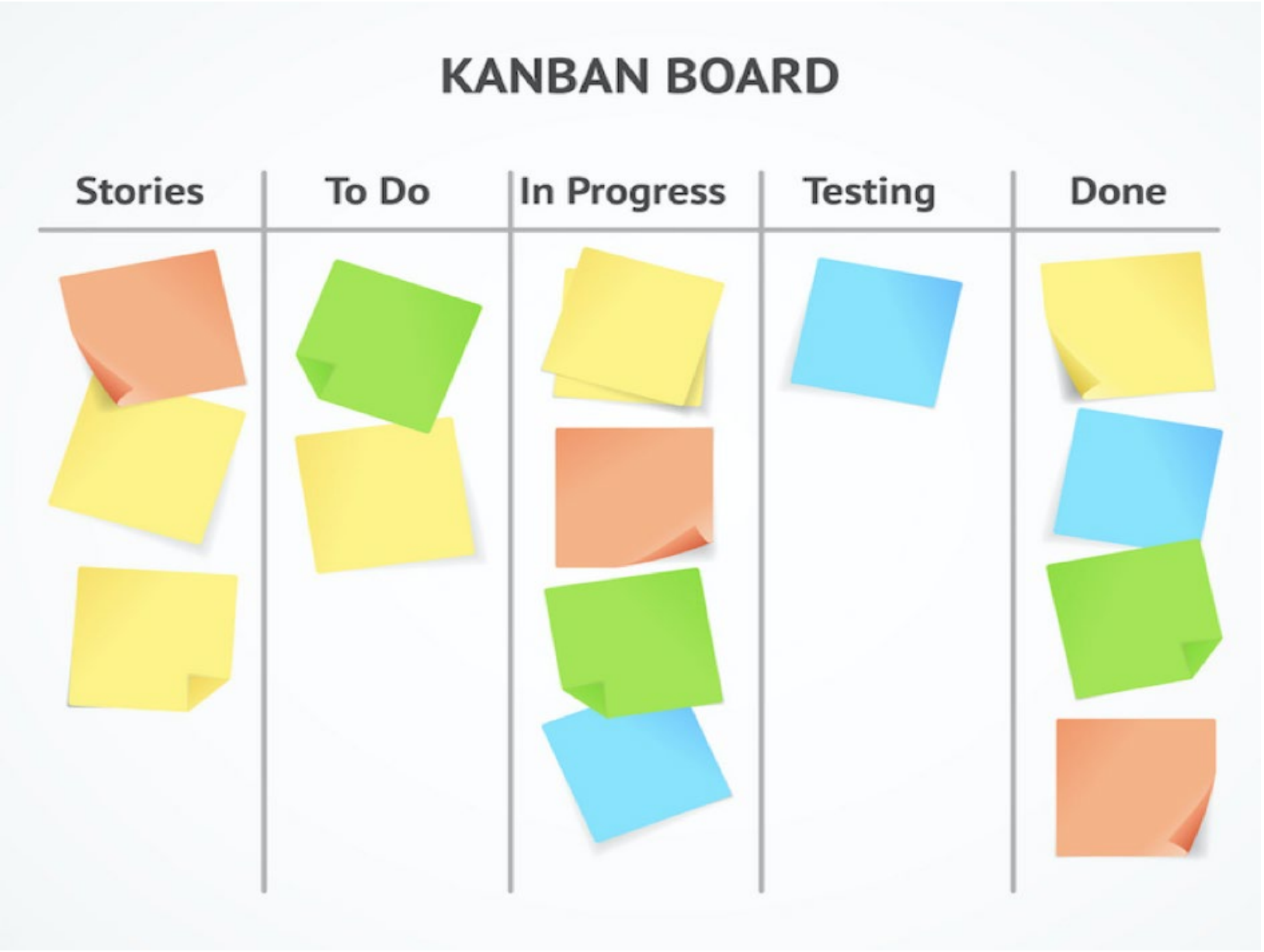
- Which of these seven categories can be applied in the classroom?
- Can you give an example?



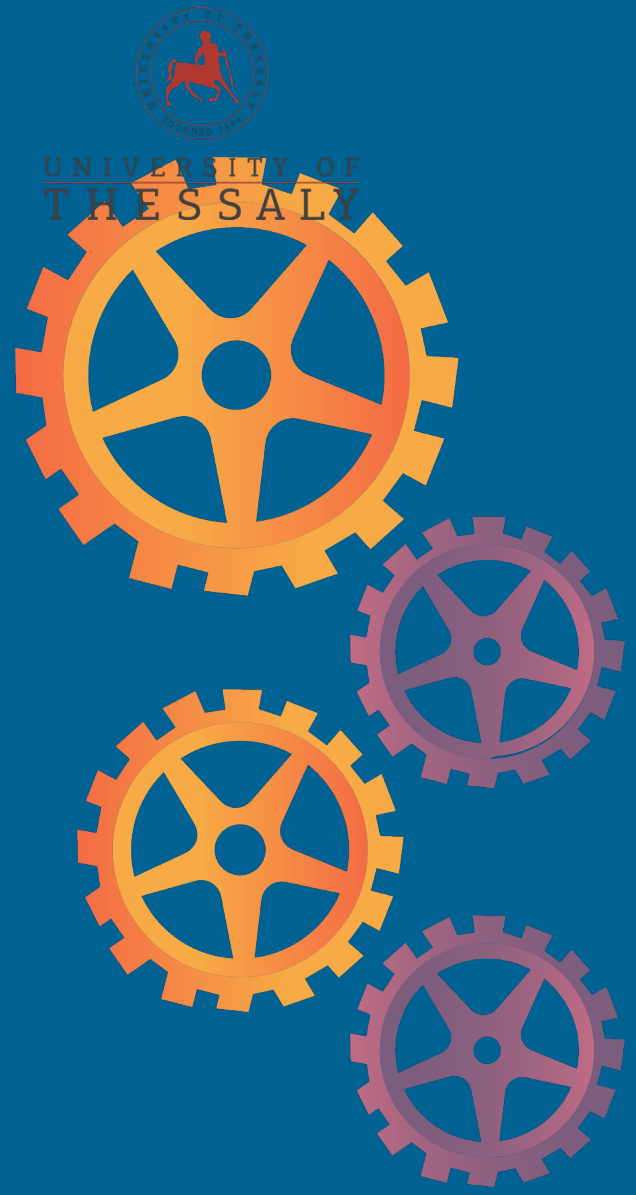


Kanban

- The word Kanban has a Japanese origin and its meaning is linked to the concept of “just in time”.
- In practice, the Kanban method is organised on a board or table (Kanban board), divided into columns, showing every flow within the software production project.
  - As the development evolves, the information contained in the table changes, and whenever a new task comes into play, a new “card” is created.
- In a Kanban team all team members need to know exactly the product’s stage development and status at any time. As that communication and transparency are critical.

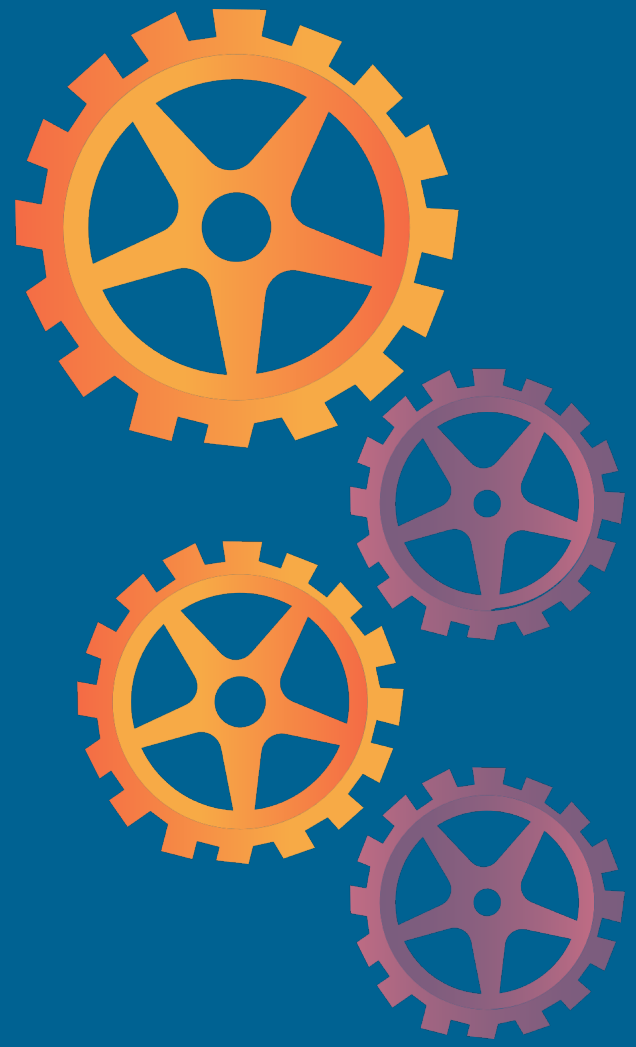


Source: https://www.bocasay.com/kanban-method-it-development-projects/



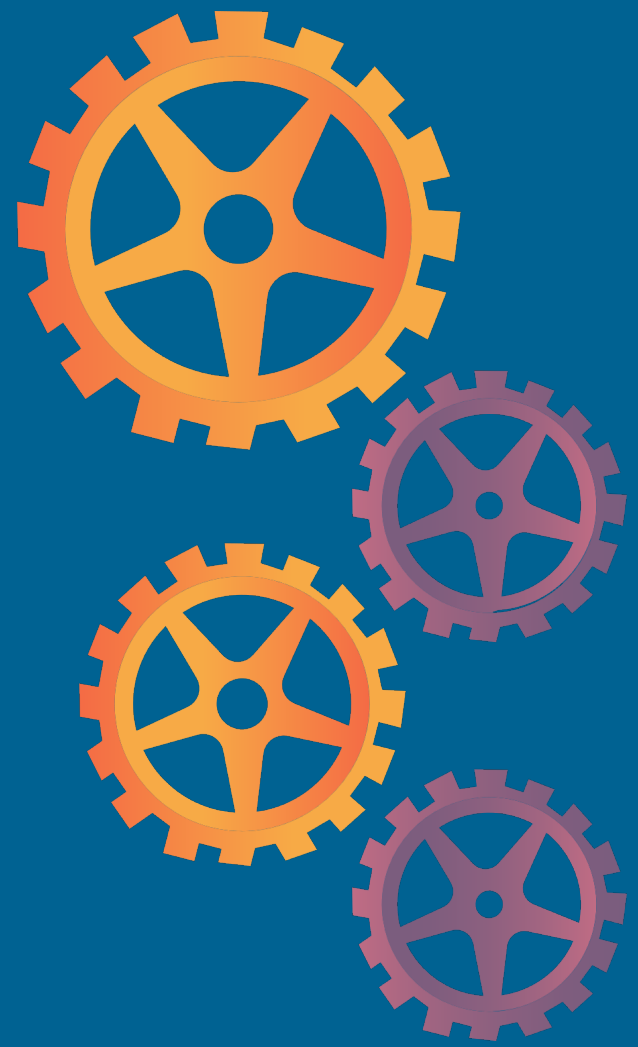
How does the Kanban team work?

- Each person on the team retains their role.
- The entire work in progress is distributed by means of labels in the columns of the visualization table.
  - The developer team deals with the tasks in the “To do” column.
  - Each task is moved from column to column, from left to right (or from its current state) until it is ready for delivery to the client, when it is placed in the “Done/completed” column.
- The aim of the Kanban method is to minimize the risk of overproduction and waste, and to reduce time and costs.
  - The visualization of the workflow allows the prioritization and tracking of the progress of the tasks.



Refereneces

- https://agilemanifesto.org/
- https://www.mountaingoatsoftware.com/agile/scrum/resources/a-reusable-scrum-presentation
- The Scrum Guide https://scrumguides.org/scrum-guide.html



Innovative Agile Project Based Learning

Co-funded by the Erasmus+ Programme of the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Thank you.

The project „Agile2Learn was financed with the support of the Erasmus+ Programme of the European Commission under the Grant No.: 2021-1-CZ01-KA220-VET-000025558

Agile2learn.eu

UNIVERSITY OF THESSALY

EPMA EUROPEAN PROJECTS & MANAGEMENT

HELLENIC OPEN UNIVERSITY

Consorzio Roma

HELLIWOOD