UNIVERSITÀ DEGLI STUDI DI TRIESTE

Class 12: Agile Toolkit

Master Course:

Data-driven Systems Engineering (ML Operations)

440MI and 305SM

Manifesto for Agile Software Development

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.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Twelve Principles of Agile Software

Agile Toolkit

- Collaboration and Communication
- Development Practices
- Planning and Tracking Tools
- Agile Frameworks and Roles
- Feedback and Customer Involvement

Agile Toolkit - Collaboration and Communication

1. Sprint Planning Meetings
2. Retrospectives
3. Backlog Refinement (Grooming)
4. Daily Scrum / Stand-up
5. Information Radiators

Agile Toolkit - Collaboration and Communication

1. Sprint Planning Meetings

Purpose:

- Define the **Sprint Goal** and select which **Product Backlog items** will be delivered during the next sprint.
- Establish a **shared understanding** of the work and clarify priorities.

Key Activities:

- Product Owner presents prioritized backlog items.
- Development Team estimates effort and commits to achievable tasks.
- Scrum Master facilitates discussion and ensures alignment with Agile principles.

Outcomes:

- A clear **Sprint Backlog** with actionable tasks.
- A **Sprint Goal** that provides focus and direction for the team.

Benefits:

- Promotes **team ownership** and accountability.
- Improves **predictability** and **collaboration** across roles.
- Aligns work with customer and business value.

Agile Toolkit - Collaboration and Communication

2. Retrospectives

Purpose:

- Encourage the team to **reflect on the previous sprint** and identify strengths, weaknesses, and opportunities for improvement.
- Foster a **culture of continuous learning** and adaptation.

Key Activities:

- Review what went **well**, what could be **improved**, and **action items** for the next sprint.
- Discuss processes, communication, and collaboration—not just technical outcomes.
- Scrum Master facilitates the session, ensuring open and constructive dialogue.

Outcomes:

- Concrete **improvement actions** for the upcoming sprint.
- Increased **team cohesion** and trust.
- Continuous enhancement of **Agile practices**.

Benefits:

- Promotes **transparency** and team **self-organization**.
- Drives **incremental improvement** in performance and morale.
- Strengthens **shared ownership** of both successes and challenges.

Agile Toolkit - Collaboration and Communication

3. Backlog Refinement

Purpose:

- Keep the **Product Backlog** organized, prioritized, and ready for future sprints.
- Ensure that backlog items are **well-defined**, **estimated**, and **aligned** with the product vision.

Key Activities:

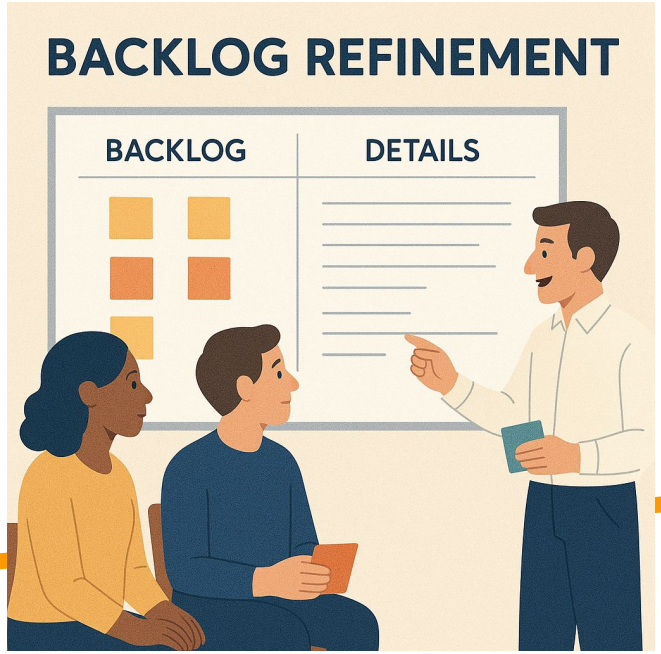
- Review and clarify user stories or backlog items.
- Add new items based on stakeholder feedback or emerging requirements.
- Reprioritize and estimate items to maintain flow and transparency.
- Collaborate across Product Owner, Developers, and Scrum Master.

Outcomes:

- A **clear, prioritized backlog** that supports effective Sprint Planning.
- Improved understanding of requirements across the whole team.
- Reduced uncertainty and surprises during sprint execution.

Benefits:

- Promotes **continuous alignment** between business goals and technical work.
- Increases **efficiency** during Sprint Planning.
- Enables **adaptive planning** as priorities evolve.



Agile Toolkit - Collaboration and Communication

4. Daily Scrum / Stand-up

Purpose:

- Provide a **short daily synchronization meeting** for the development team.
- Identify progress, obstacles, and priorities to maintain sprint focus.

Key Activities:

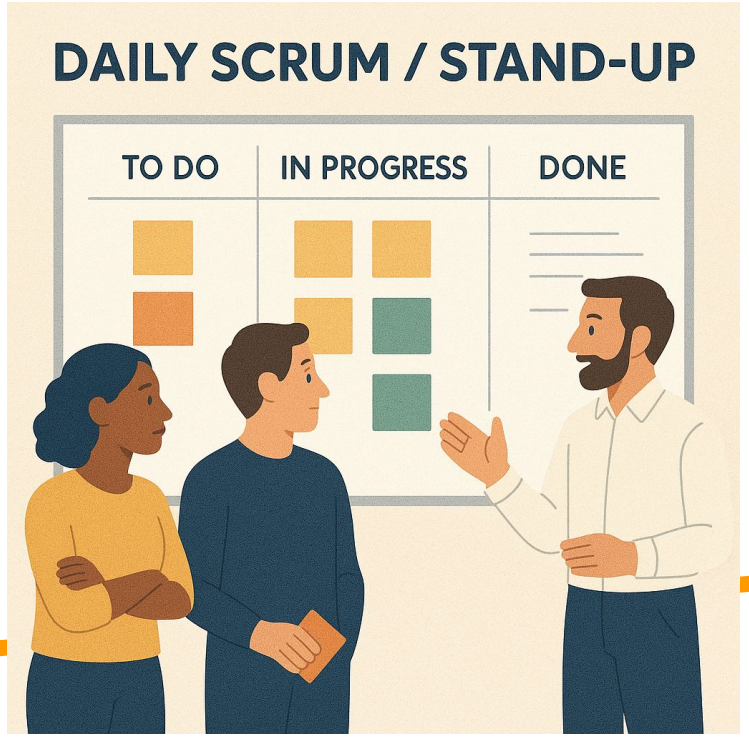
- Each team member answers three questions:
  1. *What did I do yesterday?*
  2. *What will I do today?*
  3. *Are there any impediments blocking my work?*
- Scrum Master observes and helps remove obstacles.
- The meeting is **time-boxed** (usually 15 minutes).

Outcomes:

- Clear understanding of **team progress** and next steps.
- Early detection of **risks or dependencies**.
- Reinforced **accountability** and shared ownership.

Benefits:

- Improves **communication** and **transparency**.
- Keeps the team aligned with the **Sprint Goal**.
- Encourages **self-organization** and proactive problem solving.



Agile Toolkit - Collaboration and Communication

5. Information Radiators

Purpose:

- Provide **visible, real-time information** about the team's progress, goals, and workflow.
- Enhance **transparency** and **collective awareness** across all stakeholders.

Key Activities:

- Use visual tools such as **Kanban boards**, **burndown charts**, or **task walls**.
- Continuously update status indicators (e.g., "To Do," "In Progress," "Done").
- Display key metrics like sprint progress, defects, or velocity in shared spaces.

Outcomes:

- A **shared understanding** of project status and performance.
- Early detection of **bottlenecks** or **delays**.
- Increased engagement and accountability within the team.

Benefits:

- Promotes **transparency** and **data-driven decisions**.
- Reduces the need for frequent status meetings.
- Encourages **continuous improvement** and **collaboration**.

Agile Toolkit - Development Practices

1. Test-Driven Development (TDD) – writing tests before code to guide development.
2. Behavior-Driven Development (BDD) – defining software behavior in plain language before coding.
3. Continuous Integration (CI) – frequently integrating and testing code to detect issues early.
4. Continuous Deployment/Delivery (CD) – automatically releasing validated code to production or staging.
5. Refactoring – continuously improving code structure without changing functionality.
6. Code Reviews / Peer Reviews – collaborative evaluation of code quality.

Agile Toolkit - Planning and Tracking

1. Kanban Boards
2. User Stories
3. Story Points
4. Burndown Charts
5. Velocity Tracking

Agile Toolkit - Planning and Tracking

1. Kanban Boards

Purpose:

- Visualize the **flow of work** and improve **process transparency**.
- Limit **work in progress (WIP)** to enhance focus and efficiency.

Key Activities:

- Create a board divided into workflow stages
  a. (e.g., *To Do* → *In Progress* → *Done*).
- Move tasks (cards) across stages as work progresses.
- Monitor WIP limits to prevent overload and identify bottlenecks.
- Continuously review and adapt the workflow for improvement.

Outcomes:

- Clear visibility of **current status** and **workflow efficiency**.
- Early detection of **delays or bottlenecks**.
- Balanced workload and improved delivery consistency.

Benefits:

- Promotes **flow-based work management**.
- Encourages **team ownership** and **continuous improvement**.
- Enhances **predictability** and reduces cycle time.

Agile Toolkit - Planning and Tracking

2. User Stories

Purpose:

- Capture requirements from the user's perspective.
- Focus on delivering **customer value** and shared understanding.

Key Activities:

- Write stories using the standard format: *As a [user], I want [goal] so that [benefit].*
- Discuss details collaboratively between Product Owner and Developers.
- Add **acceptance criteria** to define when a story is complete.
- Estimate effort and prioritize based on value and complexity.

Outcomes:

- Clear, testable, and user-centered requirements.
- Improved alignment between business goals and development tasks.
- Shared understanding of purpose and scope.

Benefits:

- Encourages **communication** and **empathy** for the user.
- Simplifies requirement management and refinement.
- Supports **incremental delivery** of working software.

Agile Toolkit - Planning and Tracking

3. Story Points

Purpose:

- Provide a **relative measure of effort or complexity** for user stories.
- Enable better **sprint planning** and **predictability** of team performance.

Key Activities:

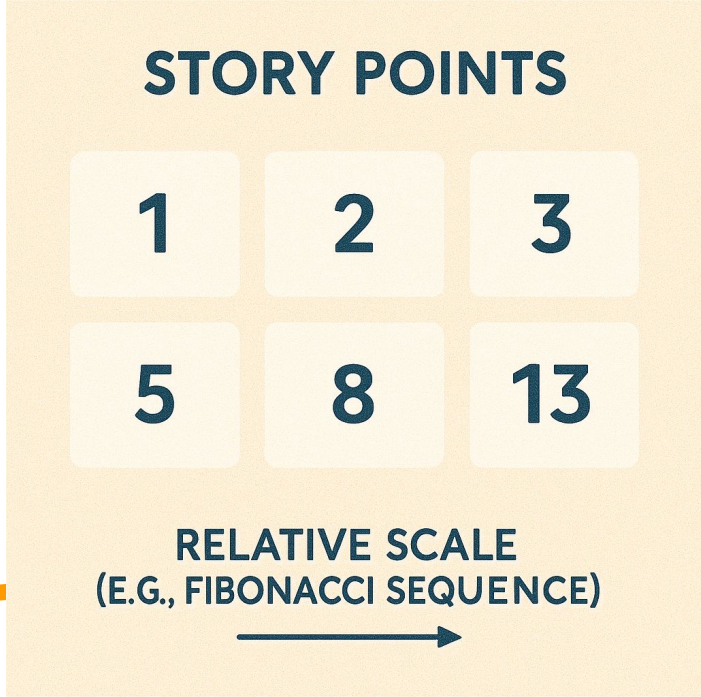
- Estimate stories using a **relative scale** (e.g., Fibonacci sequence: 1, 2, 3, 5, 8, 13...).
- Compare each story to a reference story to ensure consistency.
- Conduct **Planning Poker** sessions for collaborative estimation.
- Refine estimates as understanding improves over time.

Outcomes:

- Shared understanding of **effort and complexity** across the team.
- Improved **sprint forecasting** and **velocity tracking**.
- Enhanced team collaboration and engagement during planning.

Benefits:

- Encourages **team consensus** rather than individual judgment.
- Reduces focus on hours and promotes **value-based estimation**.
- Supports **continuous improvement** in estimating accuracy.



Agile Toolkit - Planning and Tracking

4. Burndown Charts

Purpose:

- Track **work progress** throughout the sprint or project.
- Visualize the **remaining effort** versus time to support better planning and adaptation.

Key Activities:

- Plot remaining work (in hours or story points) on the vertical axis and time on the horizontal axis.
- Update the chart daily to reflect task completion.
- Compare actual progress against the **ideal burndown line**.
- Use trends to identify potential **delays** or **scope changes** early.

Outcomes:

- Clear visualization of **team progress** toward the sprint goal.
- Early detection of **performance deviations**.
- Data-driven discussions during stand-ups and retrospectives.

Benefits:

- Enhances **transparency** and **accountability**.
- Supports **adaptive planning** and **forecasting**.
- Reinforces **continuous improvement** through feedback on team pace.

Agile Toolkit - Planning and Tracking

4. Velocity Tracking

Purpose:

- Measure the **amount of work completed** by the team in each sprint.
- Help forecast **future performance** and plan upcoming sprints realistically.

Key Activities:

- Calculate velocity as the total number of **story points completed** in a sprint.
- Track and visualize velocity across multiple sprints.
- Use historical averages to plan future sprint capacity.
- Adjust workload based on trends and team improvements.

Outcomes:

- Data-driven **capacity planning** and **sprint forecasting**.
- Improved understanding of **team performance** over time.
- Identification of **process inefficiencies** or external blockers.

Benefits:

- Enhances **predictability** of delivery.
- Encourages **continuous improvement** and realistic goal setting.
- Builds **trust** with stakeholders through consistent performance metrics.

Agile Toolkit - Agile Frameworks and Roles

1. Scrum Framework
2. Lean Practices
3. DevOps

Agile Toolkit - Agile Frameworks and Roles

1. Scrum Framework
   a. A lightweight **Agile framework** for managing complex projects.
   b. Based on **empirical process control** (transparency, inspection, adaptation).
   c. Roles: **Product Owner**, **Scrum Master**, **Development Team**.
   d. Events: **Sprint**, **Planning**, **Daily Scrum**, **Review**, **Retrospective**.
   e. Artifacts: **Product Backlog**, **Sprint Backlog**, **Increment**.
   f. Promotes **iterative delivery**, **collaboration**, and **continuous feedback**.
2. Lean Practices
   a. Originated from manufacturing principles focused on **eliminating waste**.
   b. Emphasizes **value flow**, **continuous improvement**, and **efficiency**.
   c. Encourages minimizing non-value-adding activities.
   d. Key principles: **Just-in-Time**, **Kaizen**, and **Respect for People**.
   e. Applied in Agile to **optimize delivery** and **reduce bottlenecks**.

Agile Toolkit - Agile Frameworks and Roles

3. DevOps

- Integrates **Development and Operations** to shorten release cycles.
- Promotes **automation**, **collaboration**, and **continuous delivery**.
- Uses tools for **Continuous Integration (CI)** and **Continuous Deployment (CD)**.
- Encourages shared responsibility for **software quality** and **stability**.
- Benefits: faster feedback, reduced downtime, and improved reliability.

Agile Toolkit - Feedback and Customer Involvement

1. Customer Demos / Reviews:

   > showing working increments to stakeholders for feedback.

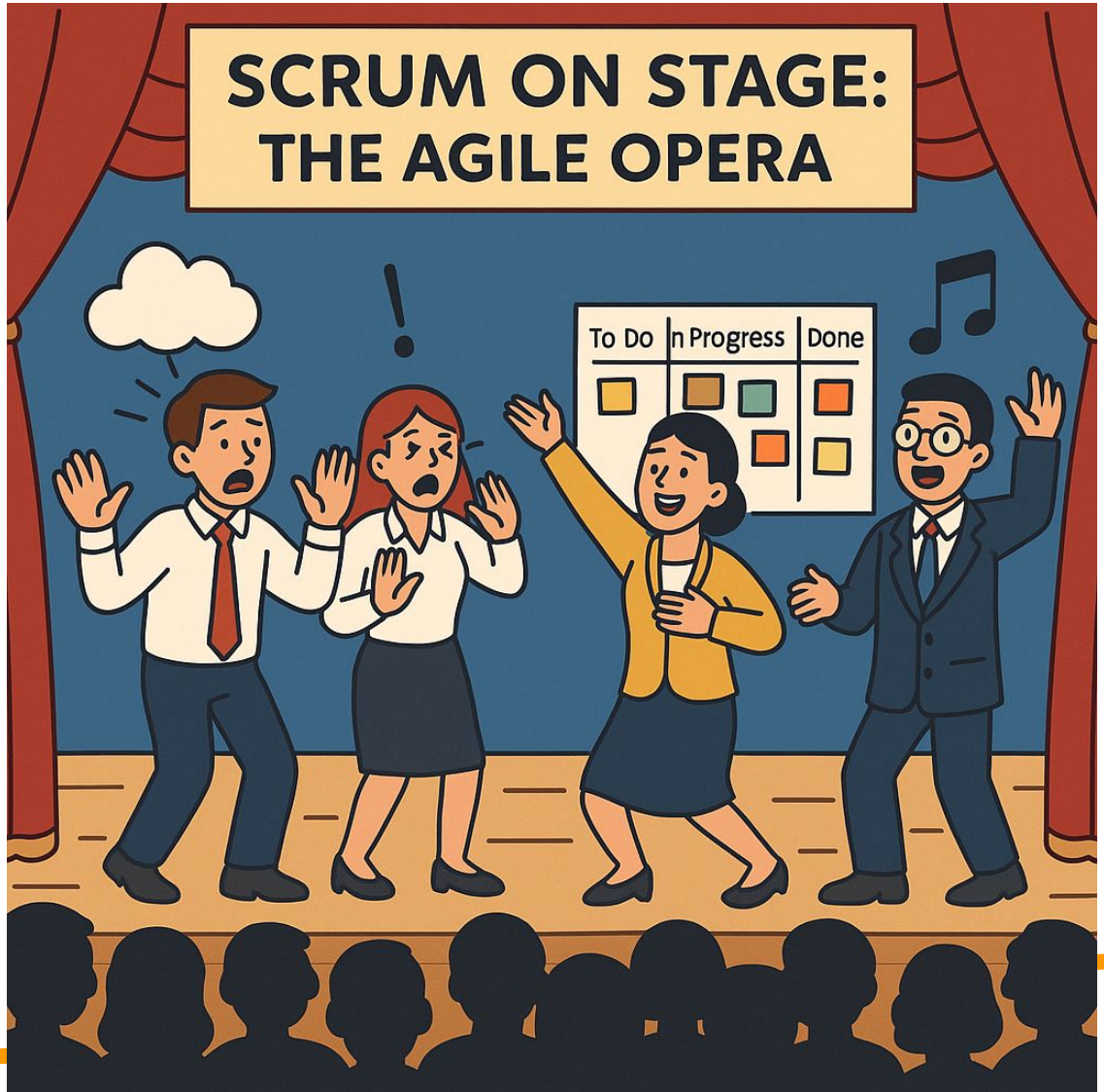
2. Minimum Viable Product (MVP):

   > early release to validate assumptions.

3. Continuous Feedback Loops

   > collecting user input for ongoing improvements.

Practical Exercise:



Thank you!