Agile Methods

[DT-0540] Metodi di sviluppo agile

Daniele Di Pompeo

daniele.dipompeo@univaq.it

UNIVERSITÀ DEGLI STUDI DELL'AQUILA A.A. 2024-2025

What's an agile method?

An agile method is a reasoned combination of some of the principles, practices, roles, and artifacts we discussed in previous lectures

Each method has its distinctive view of software development

Each method has its one Big Idea and consists of many small ideas

Examples of agile methods are:

- Extreme Programming
- Lean Software (with its Kanban variant)
- Scrum
- Spotify Model

Extreme Programming

What is Extreme Programming (XP)?

Definition:

- XP is an Agile software development methodology focused on improving software quality and responsiveness to changing requirements.
- It emphasizes frequent releases, teamwork, and continuous feedback.

Why XP?

- Suitable for rapidly changing environments.
- Encourages collaboration between developers, customers, and stakeholders.

Core Values of XP

1. **Communication:**
   - Encourages open and continuous communication within the team.
   - Regular feedback between developers and customers.
2. **Simplicity:**
   - Focus on designing and building the simplest solution that works.

Core Values of XP

3. **Feedback:**
   - Continuous feedback from the system and customer helps refine the process and product.
4. **Courage:**
   - Embrace changes and be ready to refactor and improve the code whenever needed.
5. **Respect:**
   - Value each team member's contributions and ideas.

Key Practices of XP

1. **Pair Programming:**
   - Two developers work together at one workstation, one writing code, the other reviewing it in real time.
2. **Test-Driven Development (TDD):**
   - Write tests before writing code to ensure functionality.
3. **Continuous Integration:**
   - Regularly integrate code into a shared repository to avoid integration issues.

Key Practices of XP

4. **Refactoring:**
    - Continuously improve and simplify the codebase without changing functionality.
5. **Small Releases:**
    - Deliver small, frequent releases to get feedback from customers.
6. **Simple Design:**
    - The simplest design that works should be used, avoiding over-engineering.

Key Practices of XP

7. **Collective Code Ownership:**

   - All team members have access to and are responsible for the entire codebase.

8. **Sustainable Pace:**

   - Work at a pace that can be maintained long-term without burnout.

Benefits of XP

- **Faster Development:** Small iterations and continuous feedback help deliver software more quickly.
- **Improved Code Quality:** Practices like TDD and refactoring improve the quality of the codebase.
- **Increased Flexibility:** XP encourages changes and adaptations to evolving requirements.
- **Better Collaboration:** Practices like pair programming and collective code ownership foster teamwork.

When to Use Extreme Programming (XP)?

Best suited for:

- Projects with frequently changing requirements.
- Small to medium-sized development teams.
- Environments where customer involvement is high and continuous.
- Projects that require a high degree of collaboration and communication.

XP Summary

Key Takeaways:

- XP is a highly flexible Agile methodology focused on quality and communication.
- It incorporates practices such as pair programming, TDD, and continuous integration.
- XP works best in dynamic environments with evolving requirements and strong customer interaction.

Remember: XP's practices can be adapted to many team sizes and project types!

Lean Software

What is Lean Agile?

Definition:

- Lean Agile is a combination of Lean principles and Agile practices, focused on delivering value to the customer efficiently by eliminating waste and improving flow.
- Originates from Lean manufacturing but adapted for software development to streamline processes.

Why Lean Agile?

- Helps reduce inefficiencies in development.
- Focuses on delivering high-quality products faster with fewer resources.

What is Lean Manufacturing?

Definition:

- Lean manufacturing is a systematic approach to minimizing waste within a manufacturing system while simultaneously maximizing productivity.
- Originated from the Toyota Production System (TPS) in the 1940s.

Goal:

- To create more value for customers with fewer resources by optimizing the flow of work and eliminating unnecessary processes.

Principles of Lean Agile

1. **Eliminate Waste:**
   - Remove anything that does not add value to the customer.
   - Examples include unnecessary features, delays, and excessive documentation.
2. **Build Quality In:**
   - Ensure quality at every stage of the development process, not just at the end.
   - Use testing, code reviews, and automation to maintain high quality.
3. **Create Knowledge:**
   - Continuous learning and improvement are central to Lean Agile.
   - Encourage experimentation and innovation.

Principles of Lean Agile

4. **Defer Commitment:**
   - Make decisions as late as possible when you have the most information.
   - Avoid locking in requirements early if they might change.
5. **Deliver Fast:**
   - Focus on delivering features quickly and iteratively.
   - Shorter cycles allow faster feedback and adjustments.
6. **Respect People:**
   - Trust teams to self-organize and make decisions.
   - Encourage collaboration and respect among all team members.

Key Practices of Lean Agile

1. **Value Stream Mapping:**
   - Visualize the flow of value through the development process.
   - Identify areas of waste and opportunities for improvement.
2. **Small Batch Sizes:**
   - Work in small, manageable increments to increase flow efficiency.
3. **Kanban:**
   - A visual tool to manage work by showing tasks in progress, completed, or in backlog.
4. **Continuous Improvement (Kaizen):**
   - Regularly assess processes and practices to find ways to improve.

Benefits of Lean Agile

- **Increased Efficiency:** By removing waste and improving flow, Lean Agile increases the speed and efficiency of development.
- **Improved Quality:** Built-in quality ensures that defects are caught early, leading to more reliable products.
- **Faster Delivery:** Lean Agile enables faster delivery of valuable features, providing frequent feedback loops.
- **Better Team Collaboration:** Encourages self-organization and respect among team members, improving communication and teamwork.

When to Use Lean Agile?

Best suited for:

- Projects with long-term goals where efficiency is critical.
- Teams that need to eliminate waste and streamline processes.
- Environments where there is a high emphasis on continuous improvement.
- Organizations aiming to deliver high-quality products in shorter cycles.

Lean Agile Summary

Key Takeaways:

- Lean Agile is focused on efficiency, quality, and delivering value quickly by eliminating waste.
- Practices like Kanban, small batch sizes, and value stream mapping help streamline processes.
- Lean Agile principles encourage continuous improvement, flexibility, and respecting people.

Remember: Lean Agile is all about maximizing value with minimal resources and effort!

SCRUM

What is Scrum?

Definition:

- Scrum is a lightweight Agile framework used to manage and control software development.
- It focuses on delivering small, workable increments of software in short cycles, called Sprints.

Why Scrum?

- Encourages collaboration, adaptability, and continuous feedback.
- Helps teams manage complex projects efficiently by breaking work into manageable pieces.

Core Principles of Scrum

1. **Empiricism:**
   - Scrum is based on the idea that knowledge comes from experience and decision-making is based on what is known.
   - Regular inspection and adaptation help the team adjust and improve.

2. **Self-Organization:**
   - Scrum teams are self-organized, meaning that they decide how best to accomplish the work.

3. **Incremental Progress:**
   - Scrum delivers work in small, iterative increments, allowing for early feedback and adaptation.

Scrum Roles

1. **Product Owner:**
   - Represents the stakeholders and customers.
   - Responsible for maximizing the value of the product and prioritizing the product backlog.
2. **Scrum Master:**
   - Facilitates the Scrum process and removes obstacles that impede progress.
   - Ensures the team follows Scrum practices and improves over time.
3. **Development Team:**
   - A cross-functional group responsible for delivering potentially shippable increments of the product at the end of each Sprint.

Scrum Meetings

1. **Sprint Planning:**
   - The team collaborates to define what can be delivered in the upcoming Sprint.
   - The Product Owner prioritizes the backlog, and the team selects the work.
2. **Daily Scrum:**
   - A short, daily meeting where the team discusses progress and identifies roadblocks.

Scrum Meetings

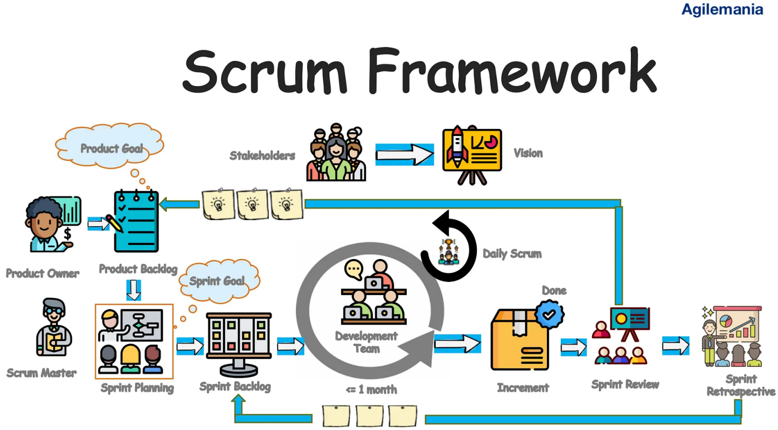
3. **Sprint Review:**
   - At the end of the Sprint, the team presents their work to stakeholders and gathers feedback.
4. **Sprint Retrospective:**
   - The team reflects on what went well, what didn't, and how they can improve in the next Sprint.

Scrum Artifacts

1. **Product Backlog:**
   - A prioritized list of everything that might be needed in the product.
   - Managed by the Product Owner and continuously updated based on feedback and changing requirements.
2. **Sprint Backlog:**
   - A list of tasks the development team plans to complete during the current Sprint.
3. **Increment:**
   - The sum of all completed product backlog items at the end of the Sprint, representing potentially shippable product functionality.

Scrum: Visual Representation

https://medium.com/agilemania/essential-elements-of-agile-scrum-d5d5d8cafe8d



Benefits of Scrum

- **Improved Collaboration:** Frequent communication within the team ensures everyone is aligned and obstacles are addressed quickly.
- **Faster Delivery:** Incremental releases allow teams to deliver functionality faster and respond to changes quickly.
- **Better Quality:** Regular reviews and feedback ensure that quality is built into the product from the start.
- **Increased Flexibility:** Scrum allows teams to adapt to changing requirements and priorities.

When to Use Scrum?

Best suited for:

- Projects with rapidly changing requirements or environments.
- Teams that need frequent feedback and collaboration with stakeholders.
- Medium to large projects where tasks can be broken down into small, manageable chunks.

Scrum Summary

Key Takeaways:

- Scrum is an Agile framework that focuses on delivering incremental value through collaboration and adaptability.
- The framework includes specific roles (Product Owner, Scrum Master, Development Team), meetings (Sprint Planning, Daily Scrum, Sprint Review, Retrospective), and artifacts (Product Backlog, Sprint Backlog, Increment).
- Scrum is well-suited for dynamic projects requiring flexibility and continuous improvement.

Remember: Scrum emphasizes teamwork, accountability, and iterative progress toward well-defined goals.

Spotify Model

Introduction to the Spotify Model

- Developed by Spotify, the music streaming company
- An approach to scaling agile methodologies
- Emphasizes:
  - Autonomy
  - Communication
  - Alignment
- Flexible and adaptable, not a rigid framework

Key Components: Squads

- Small, cross-functional teams
- Similar to Scrum teams
- Work autonomously on specific features or products
- Typically 6-12 people
- Have all skills needed to design, develop, test, and release their part of the product

Key Components: Tribes

- Collections of squads working in related areas
- Typically up to 100 people
- Share common objectives and work on related features
- Led by a Tribe Lead
- Promote collaboration between squads

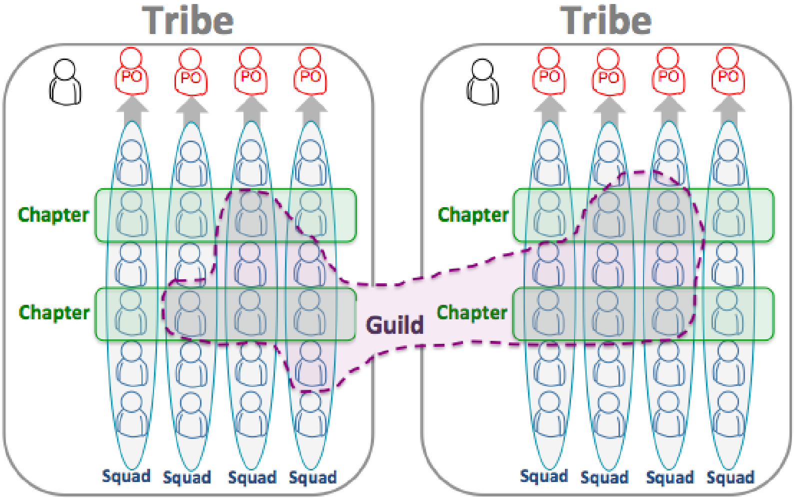
Key Components: Chapters

- Groups of people with similar skills across different squads
- Example: All testers in a tribe form a chapter
- Led by a Chapter Lead (usually a line manager)
- Foster skill development
- Share best practices within their area of expertise

Key Components: Guilds

- Informal communities of interest
- Span the entire organization
- Allow people to share knowledge on specific topics
- Voluntary participation
- Examples: Web Technology Guild, Agile Coach Guild

Spotify Model: Visual Representation



Benefits of the Spotify Model

- Balances autonomy and alignment
- Promotes innovation and creativity
- Enhances knowledge sharing across the organization
- Improves employee satisfaction through autonomy
- Allows for flexible team structures
- Scales agile practices to large organizations

Challenges and Considerations

- Not a one-size-fits-all solution
- Requires a culture of trust and employee empowerment
- Can be complex to implement in traditional hierarchical structures
- Needs strong communication to maintain alignment
- Requires continuous adaptation and improvement

Conclusion

- The Spotify Model offers a flexible approach to scaling agile
- Key components: Squads, Tribes, Chapters, and Guilds
- Balances team autonomy with organizational alignment
- Encourages innovation, knowledge sharing, and employee satisfaction
- Remember: Adapt the model to fit your organization's needs

Agile Methods

[DT-0540] Metodi di sviluppo agile

Daniele Di Pompeo

daniele.dipompeo@univaq.it

UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

A.A. 2024-2025