# Software development methodologies

## Agile Methodology

prof. Leonardo Essam Dei Rossi

ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

# Indice

1. What is the Agile methodology?
   - Key Takeaways
2. Why choose Agile?
3. The phases of Agile
   - Phase #1: Requirements
   - Phase #2: Design
   - Phase #3: Development
   - Phase #4: Testing
   - Phase #5: Deployment
   - Phase #6: Review
4. Agile vs. Waterfall

# Indice

1. What is the Agile methodology?
    - Key Takeaways
2. Why choose Agile?
3. The phases of Agile
    - Phase #1: Requirements
    - Phase #2: Design
    - Phase #3: Development
    - Phase #4: Testing
    - Phase #5: Deployment
    - Phase #6: Review
4. Agile vs. Waterfall

# What is the Agile methodology? (1)

The Agile methodology is an approach that divides work into phases, emphasizing continuous delivery and improvement. Agile benefits teams by enabling adaptive planning, rapid execution, and ongoing evaluation, leading to more responsive and successful outcomes.

The traditional ”waterfall” approach has each team work on a project separately, then hand it off to the next group. On the other hand, agile utilizes collaborative, cross-functional teams, emphasizes open communication, teamwork, adaptability, and trust.

# What is the Agile methodology? (1)

The Agile methodology is an approach that divides work into phases, emphasizing continuous delivery and improvement. Agile benefits teams by enabling adaptive planning, rapid execution, and ongoing evaluation, leading to more responsive and successful outcomes.

The traditional ”waterfall” approach has each team work on a project separately, then hand it off to the next group. On the other hand, agile utilizes collaborative, cross-functional teams, emphasizes open communication, teamwork, adaptability, and trust.

# What is the Agile methodology? (2)

Although the project lead or product owner typically prioritizes the work to be delivered, the team takes the lead on deciding how the work will get done, self-organizing around granular tasks and assignments. Agile isn't defined by a set of ceremonies or specific development techniques.

Instead, agile is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement. The original Agile Manifesto didn't prescribe two-week iterations or an ideal team size.

# What is the Agile methodology? (2)

Although the project lead or product owner typically prioritizes the work to be delivered, the team takes the lead on deciding how the work will get done, self-organizing around granular tasks and assignments. Agile isn't defined by a set of ceremonies or specific development techniques.

Instead, agile is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement. The original Agile Manifesto didn't prescribe two-week iterations or an ideal team size.

# Key Takeaways

- Agile is a flexible, iterative approach to project management that emphasizes collaboration, continuous delivery, and adaptability;
- It values people, customer feedback, and working solutions over rigid processes and documentation;
- Teams tailor Agile practices to their needs, blending frameworks like Scrum and Kanban for optimal results;
- Experiment with Agile practices and regularly review their effectiveness to foster continuous improvement and team collaboration.

# Key Takeaways

- Agile is a flexible, iterative approach to project management that emphasizes collaboration, continuous delivery, and adaptability;
- It values people, customer feedback, and working solutions over rigid processes and documentation;
- Teams tailor Agile practices to their needs, blending frameworks like Scrum and Kanban for optimal results;
- Experiment with Agile practices and regularly review their effectiveness to foster continuous improvement and team collaboration.

# Key Takeaways

- Agile is a flexible, iterative approach to project management that emphasizes collaboration, continuous delivery, and adaptability;
- It values people, customer feedback, and working solutions over rigid processes and documentation;
- Teams tailor Agile practices to their needs, blending frameworks like Scrum and Kanban for optimal results;
- Experiment with Agile practices and regularly review their effectiveness to foster continuous improvement and team collaboration.

# Key Takeaways

- Agile is a flexible, iterative approach to project management that emphasizes collaboration, continuous delivery, and adaptability;
- It values people, customer feedback, and working solutions over rigid processes and documentation;
- Teams tailor Agile practices to their needs, blending frameworks like Scrum and Kanban for optimal results;
- Experiment with Agile practices and regularly review their effectiveness to foster continuous improvement and team collaboration.

# Indice

1. What is the Agile methodology?
   - Key Takeaways
2. Why choose Agile?
3. The phases of Agile
   - Phase #1: Requirements
   - Phase #2: Design
   - Phase #3: Development
   - Phase #4: Testing
   - Phase #5: Deployment
   - Phase #6: Review
4. Agile vs. Waterfall

# Why choose agile? (1)

Teams choose agile methodologies so they can respond to changes in the marketplace or customer feedback quickly, without derailing a year's worth of plans. "Just enough" planning and shipping in small, frequent increments lets your team gather feedback on each change and integrate it into future plans at minimal cost.

But it's not just a numbers game-first and foremost, it's about people. As described by the Agile Manifesto, authentic human interactions are more important than rigid processes.

# Why choose agile? (1)

Teams choose agile methodologies so they can respond to changes in the marketplace or customer feedback quickly, without derailing a year's worth of plans. "Just enough" planning and shipping in small, frequent increments lets your team gather feedback on each change and integrate it into future plans at minimal cost.

But it's not just a numbers game-first and foremost, it's about people. As described by the Agile Manifesto, authentic human interactions are more important than rigid processes.

# Why choose agile? (2)

Collaborating with customers and teammates is more important than predefined arrangements. Delivering a working solution to the customer's problem is more important than providing hyper-detailed documentation.

An agile team unites under a shared vision, then brings it to life the way they know is best. Each team sets its own standards for quality, usability, and completeness.

# Why choose agile? (2)

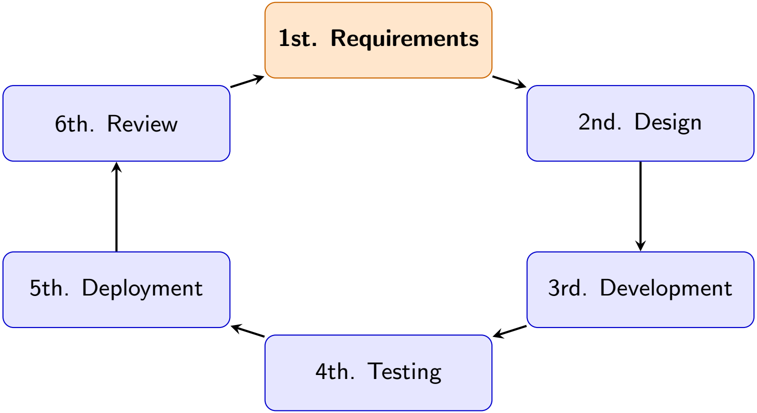
Collaborating with customers and teammates is more important than predefined arrangements. Delivering a working solution to the customer's problem is more important than providing hyper-detailed documentation.

An agile team unites under a shared vision, then brings it to life the way they know is best. Each team sets its own standards for quality, usability, and completeness.

# Indice

1. What is the Agile methodology?
    - Key Takeaways
2. Why choose Agile?
3. The phases of Agile
    - Phase #1: Requirements
    - Phase #2: Design
    - Phase #3: Development
    - Phase #4: Testing
    - Phase #5: Deployment
    - Phase #6: Review
4. Agile vs. Waterfall

# The phases of Agile



**1st. Requirements**

2nd. Design

3rd. Development

4th. Testing

5th. Deployment

6th. Review

# Phase #1: Requirements

In the Waterfall model, requirements are a ”signed-and-sealed” contract. In Agile, they are a conversation.

We move away from rigid documentation toward a **Product Backlog** – a prioritized list of user needs.

This phase is about understanding the intent behind a feature rather than just a checklist of technical specs. By focusing on ”User Stories”, the team remains outcome-oriented, allowing the requirements to evolve as market conditions or user feedback change.

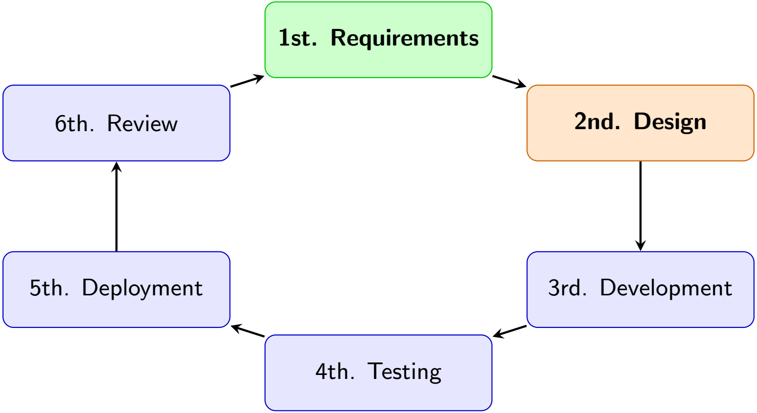
# Phase #1: Requirements

In the Waterfall model, requirements are a "signed-and-sealed" contract. In Agile, they are a conversation.

We move away from rigid documentation toward a **Product Backlog** – a prioritized list of user needs.

This phase is about understanding the intent behind a feature rather than just a checklist of technical specs. By focusing on "User Stories", the team remains outcome-oriented, allowing the requirements to evolve as market conditions or user feedback change.

# The phases of Agile



**1st. Requirements**

**2nd. Design**

3rd. Development

4th. Testing

5th. Deployment

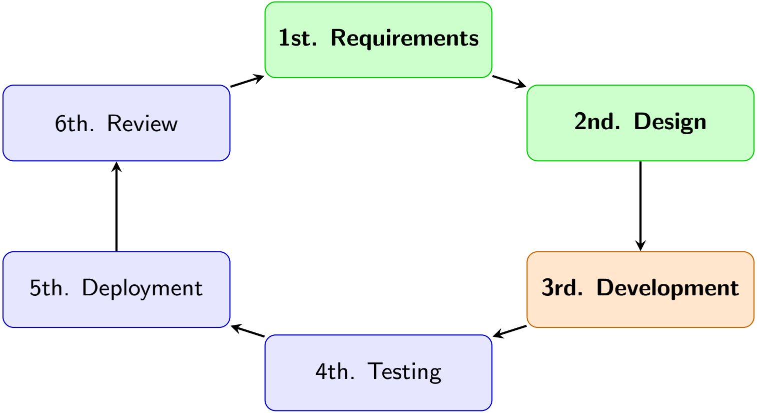
6th. Review

# Phase #2: Design

**Agile design rejects the "Big Design Up Front" (BDUF) philosophy.**

Instead, it favors evolutionary architecture. The team spends enough time to ensure the system is scalable and user-friendly, but they don't try to solve every problem that might occur three years from now. The goal is to create a flexible blueprint that allows for technical pivots without needing to scrap months of planning.

# The phases of Agile



# Phase #3: Development

**This is the ”engine room” of the cycle.**

Development is broken into short, focused bursts (*sprints*). Rather than building the entire system at once, developers focus on creating a **Potentially Shippable Product Increment** (PSPI). This ensures that even if the project were halted tomorrow, the team would still have a functional piece of software to show for their efforts.

It’s an approach that prioritizes ”working software over comprehensive documentation”.

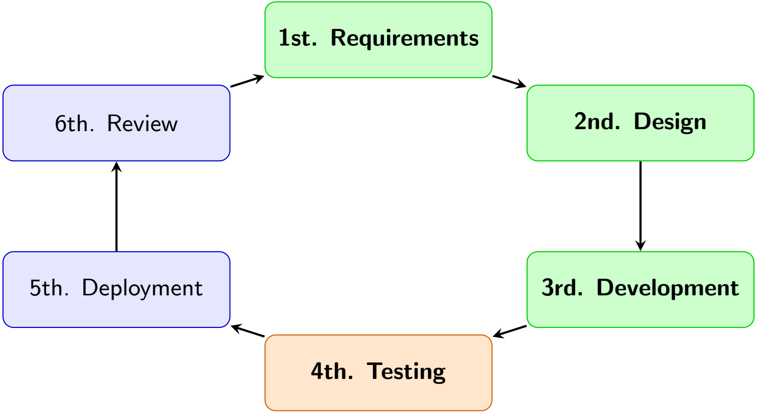
# Phase #3: Development

**This is the "engine room" of the cycle.**

Development is broken into short, focused bursts (*sprints*). Rather than building the entire system at once, developers focus on creating a **Potentially Shippable Product Increment** (PSPI). This ensures that even if the project were halted tomorrow, the team would still have a functional piece of software to show for their efforts.

It's an approach that prioritizes "working software over comprehensive documentation".

# The phases of Agile



**1st. Requirements**

**2nd. Design**

**3rd. Development**

**4th. Testing**

5th. Deployment

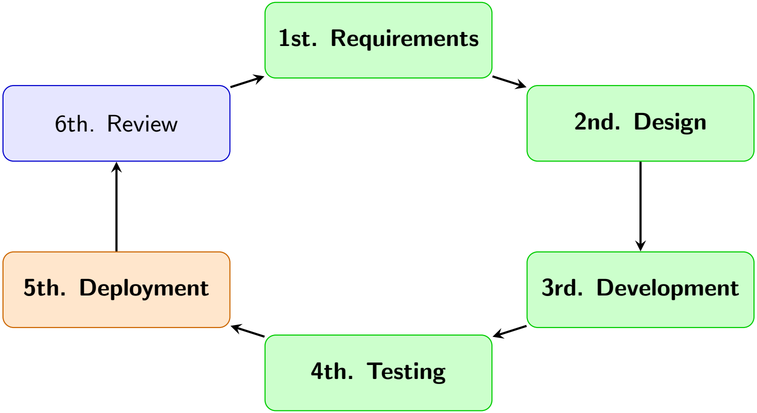
6th. Review

# Phase #4: Testing

**Testing in Agile is not a ”gate” at the end of a project, it is woven into the very fabric of development.**

By implementing *Continuous Testing* [1] and automated suites, the team ensures that new features don’t break existing functionality. This shift-left approach means bugs are caught while the code is still fresh in the developer’s mind, drastically reducing the cost and time of repairs.

# The phases of Agile

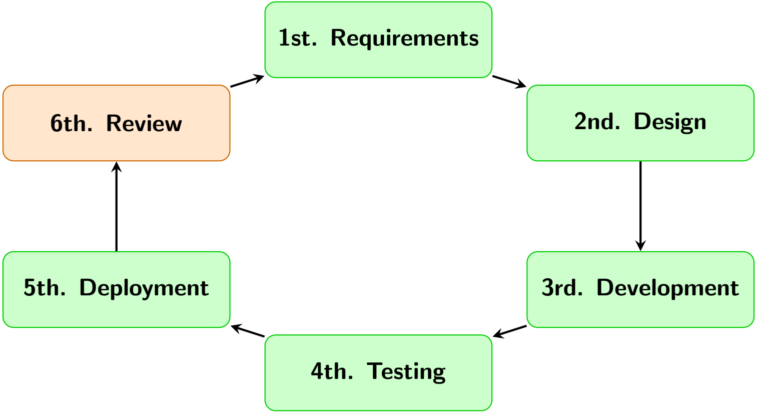


# Phase #5: Deployment

**The deployment phase is where the ”rubber meets the road”.**

In a mature Agile environment, this often utilizes *Continuous Delivery* [2] pipelines. The objective is to get the software into the hands of actual users as quickly and safely as possible. This allows the business to realize value immediately and provides the team with real-world data on how the feature is actually being used.

# The phases of Agile



1st. Requirements

2nd. Design

3rd. Development

4th. Testing

5th. Deployment

6th. Review

# Phase #6: Review

The final phase is perhaps the most critical for the Agile philosophy of ”inspect and adapt”:

- **The Sprint Review:**
  brings in stakeholders to validate that the team is building the right thing;
- **The Retrospective:**
  allows the team to look inward to ensure they are building it the right way.

This isn’t just a wrap-up; it’s the bridge to the next cycle, ensuring that every mistake becomes a lesson for the next iteration.

# Phase #6: Review

The final phase is perhaps the most critical for the Agile philosophy of "inspect and adapt":

- **The Sprint Review:**
  brings in stakeholders to validate that the team is building the right thing;
- **The Retrospective:**
  allows the team to look inward to ensure they are building it the right way.

This isn't just a wrap-up; it's the bridge to the next cycle, ensuring that every mistake becomes a lesson for the next iteration.

# Indice

1. What is the Agile methodology?
   - Key Takeaways
2. Why choose Agile?
3. The phases of Agile
   - Phase #1: Requirements
   - Phase #2: Design
   - Phase #3: Development
   - Phase #4: Testing
   - Phase #5: Deployment
   - Phase #6: Review
4. Agile vs. Waterfall

# Agile vs. Waterfall (1)

Agile was first adopted by software teams, who moved from the traditional, sequential waterfall approach to a method that garnered consistent feedback and adjustment throughout the development lifecycle.

Agile project management takes an iterative approach to development by creating several incremental steps with regular feedback intervals. This promotes adaptability since a team can adjust throughout the product development process, rather than being confined to a linear path. It also allows for regular, high-impact releases that enable teams to deliver a series of wins over time.

# Agile vs. Waterfall (1)

Agile was first adopted by software teams, who moved from the traditional, sequential waterfall approach to a method that garnered consistent feedback and adjustment throughout the development lifecycle.

Agile project management takes an iterative approach to development by creating several incremental steps with regular feedback intervals. This promotes adaptability since a team can adjust throughout the product development process, rather than being confined to a linear path. It also allows for regular, high-impact releases that enable teams to deliver a series of wins over time.

# Agile vs. Waterfall (2)

Iterative releases unlock multiple opportunities for a team to:

- Adapt to changing circumstances from newly discovered requirements to a blocked piece of work;
- Gather feedback from stakeholders during the process and iterate responsively without the stress of a final delivery deadline;
- Build relationships and connections across roles that make it easier for people to connect and communicate effectively.

Agile allows teams to be more resilient to changes that inevitably occur during a project.

# Agile vs. Waterfall (2)

Iterative releases unlock multiple opportunities for a team to:

- Adapt to changing circumstances from newly discovered requirements to a blocked piece of work;
- Gather feedback from stakeholders during the process and iterate responsively without the stress of a final delivery deadline;
- Build relationships and connections across roles that make it easier for people to connect and communicate effectively.

Agile allows teams to be more resilient to changes that inevitably occur during a project.

# Agile vs. Waterfall (2)

Iterative releases unlock multiple opportunities for a team to:

- Adapt to changing circumstances from newly discovered requirements to a blocked piece of work;
- Gather feedback from stakeholders during the process and iterate responsively without the stress of a final delivery deadline;
- Build relationships and connections across roles that make it easier for people to connect and communicate effectively.

Agile allows teams to be more resilient to changes that inevitably occur during a project.

# Agile vs. Waterfall (2)

Iterative releases unlock multiple opportunities for a team to:

- Adapt to changing circumstances from newly discovered requirements to a blocked piece of work;
- Gather feedback from stakeholders during the process and iterate responsively without the stress of a final delivery deadline;
- Build relationships and connections across roles that make it easier for people to connect and communicate effectively.

Agile allows teams to be more resilient to changes that inevitably occur during a project.

# Riferimenti e approfondimenti

[1] Wikipedia contributors, *Continuous testing — Wikipedia, The Free Encyclopedia*, [Online; accessed 25-February-2026], 2025. indirizzo: https://en.wikipedia.org/w/index.php?title=Continuous_testing&oldid=1329392723.

[2] Wikipedia contributors, *Continuous delivery — Wikipedia, The Free Encyclopedia*, [Online; accessed 25-February-2026], 2025. indirizzo: https://en.wikipedia.org/w/index.php?title=Continuous_delivery&oldid=1322843384.