Agile Methodology

---

1. Introduction

- Agile is a **software development methodology** that promotes **flexibility, iterative progress, and customer collaboration**.
- It was introduced to **overcome the limitations of traditional (Waterfall) software development**.
- Agile development follows an **incremental and iterative** approach, where software is developed in **small, manageable cycles**.
- Agile emphasizes **continuous feedback, adaptability, and customer satisfaction** over rigid planning.

---

2. Agile Manifesto – Core Values

The **Agile Manifesto (2001)** defines four key values:

1. **Individuals and interactions** over processes and tools.
    - Encourages open communication, teamwork, and collaboration.
2. **Working software** over comprehensive documentation.
    - Focuses on delivering functional software instead of extensive documentation.
3. **Customer collaboration** over contract negotiation.
    - Customers are involved throughout the development process.
4. **Responding to change** over following a plan.
    - Agile welcomes changes in requirements, even late in development.

---

3. Agile Principles

The Agile Manifesto also defines **12 principles**:

1. **Customer satisfaction** is the highest priority.
2. Welcome **changing requirements**, even in late development.
3. Deliver **working software frequently** (short iterations of 1-4 weeks).
4. Close collaboration between **business stakeholders and developers**.
5. **Motivated individuals** should be trusted to get the job done.
6. Face-to-face communication is the **most effective** way to convey information.
7. Working software is the **primary measure of progress**.
8. Maintain a **sustainable development pace** (avoid burnout).
9. **Continuous attention** to technical excellence and good design.
10. **Simplicity** – focus on only essential work.
11. **Self-organizing teams** produce the best results.
12. Teams should **reflect and improve** at regular intervals.

4. Agile Methodologies (Frameworks)

Agile is an umbrella term for multiple frameworks, including:

4.1 Scrum

- **Most widely used Agile framework.**
- Software is developed in fixed-length iterations called **Sprints** (usually 2-4 weeks).
- **Roles in Scrum:**
  - **Scrum Master:** Facilitates Scrum practices and removes blockers.
  - **Product Owner:** Defines product requirements and backlog.
  - **Development Team:** Builds, tests, and delivers the product.
- **Key Scrum Events:**
  - **Sprint Planning:** Team plans tasks for the sprint.
  - **Daily Standup:** 15-minute meeting to track progress.
  - **Sprint Review:** Demonstration of completed work.
  - **Sprint Retrospective:** Reflects on improvements for the next sprint.

4.2 Kanban

- A **visual workflow management method** using a **Kanban board**.
- Focuses on **continuous delivery** without fixed-length sprints.
- Limits **Work in Progress (WIP)** to optimize flow.
- Helps teams manage ongoing projects efficiently.

4.3 Extreme Programming (XP)

- Focuses on **engineering best practices** like:
  - **Test-Driven Development (TDD)** – Writing tests before coding.
  - **Pair Programming** – Two developers working on the same code.
  - **Continuous Integration (CI)** – Frequent code integration and testing.
- Suitable for teams requiring **high-quality code and frequent releases**.

4.4 Lean Development

- Inspired by **Toyota's Lean Manufacturing principles**.
- Focuses on **eliminating waste**, improving flow, and delivering value faster.
- Encourages **minimal documentation, fast feedback, and customer involvement**.

4.5 Feature-Driven Development (FDD)

- Focuses on developing **features in short iterations**.
- Follows a structured approach with:
  - Domain modeling.
  - Feature planning.
  - Iterative development cycles.

4.6 Dynamic Systems Development Method (DSDM)

- Prioritizes **business needs and rapid delivery**.
- Uses **MoSCoW prioritization** (Must have, Should have, Could have, Won't have).
- Ensures **customer involvement** at every stage.

---

5. Agile vs. Waterfall

Feature	Agile	Waterfall
Development Approach	Iterative & Incremental	Sequential
Flexibility	High (welcomes changes)	Low (fixed requirements)
Testing	Continuous	Performed after development
Customer Involvement	High	Low
Risk Management	Low risk (frequent reviews)	High risk (late testing)
Delivery	Frequent small releases	Single final release

---

6. Agile Lifecycle (Workflow)

1. **Concept & Requirement Gathering** – Define high-level business needs.
2. **Product Backlog Creation** – List features & tasks prioritized by importance.
3. **Sprint Planning** – Select tasks for the current iteration.
4. **Sprint Execution & Development** – Coding, testing, and feature implementation.
5. **Daily Stand-up Meetings** – Team updates progress and discusses roadblocks.
6. **Sprint Review & Demonstration** – Present completed work to stakeholders.
7. **Sprint Retrospective** – Discuss what went well and what to improve.
8. **Next Sprint Begins** – Repeat the cycle.

---

7. Benefits of Agile

- **Faster Time-to-Market** – Delivers usable software in early iterations.
- **Higher Customer Satisfaction** – Frequent feedback ensures alignment with customer needs.
- **Better Quality** – Continuous testing and quick bug fixes.
- **More Flexibility** – Easily accommodates changing requirements.
- **Stronger Team Collaboration** – Encourages daily communication.
- **Reduced Risks** – Early defect detection and faster resolution.

---

8. Challenges in Agile

- **Frequent Requirement Changes** – Can be hard to manage in large projects.
- **High Customer Involvement Needed** – Not always feasible.
- **Difficult to Estimate Costs & Time** – Due to evolving scope.
- **Requires Skilled Teams** – Self-organized teams must be capable of making decisions.

---

9. When to Use Agile?

✔ Projects with **changing requirements**.
✔ When **quick delivery** is needed.
✔ When **customer involvement** is high.
✔ For **complex and innovative** software development.
✔ When a **flexible and adaptive** approach is required.

---

10. Conclusion

- Agile is a **modern, flexible, and iterative approach** to software development.
- It **prioritizes customer satisfaction, collaboration, and rapid delivery**.
- Agile frameworks like **Scrum, Kanban, and XP** help teams improve **efficiency and quality**.
- Though Agile has challenges, it is widely adopted in **IT, finance, healthcare, and e-commerce** industries.

By mastering Agile, teams can develop **high-quality software** while adapting to **changing business needs** efficiently.