

Agile Development Process


Introduction

Agile Development is a modern software development methodology that emphasizes:

- Flexibility in responding to change
- Customer collaboration over strict contracts
- Rapid delivery of working software
- Iterative and incremental development

It originated from the Agile Manifesto (2001), which highlighted 4 core values:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a fixed plan

 Agile is not a single process model but a set of principles that guide development. The most popular framework of Agile is Scrum, but others like Kanban, Extreme Programming (XP), and Lean also exist.

Agile Development Process (Detailed Steps)

Unlike traditional Waterfall, Agile does not follow a rigid "step-by-step" sequence. Instead, it works in short, repeatable cycles called Iterations or Sprints (usually 1–4 weeks). Each iteration produces a usable increment of software.

1. Requirement Gathering and Backlog Creation

- Requirements are expressed as user stories (short, simple descriptions of a feature from the end-user's perspective).
 - All user stories are collected in a Product Backlog (a prioritized list of features, bugs, enhancements).
 - The Product Owner (in Scrum) decides priority.
-

2. Sprint / Iteration Planning

- The development team selects items from the product backlog to work on in this sprint.
 - These selected items become the Sprint Backlog. Planning is lightweight → Agile does not lock all requirements upfront.
-

3. Iterative Development (Design + Coding + Testing)

- The team works on designing, coding, and testing the selected features.

- Agile promotes practices like:
 - Pair Programming (two developers code together). Test-Driven
 - Development (TDD) (tests are written before code). Continuous
 - Integration (CI) (code is integrated and tested frequently).
 - At the end of the sprint, the team delivers a working software increment.
-

4. Daily Stand-Up Meetings (Scrum Meetings)

- Every day, the team meets for 15 minutes to answer three questions:
 1. What did I do yesterday?
 2. What will I do today?
 3. What obstacles are blocking my work?
 - This keeps everyone aligned and problems visible.
-

5. Sprint Review (Customer Feedback)

- At the end of the sprint, the team demonstrates the working product to stakeholders/customers.
 - Customers give feedback, and new requirements may be added to the backlog.
 - This ensures the product is always moving in the right direction.
-

6. Retrospective (Process Improvement)

- The team reflects on the sprint:
 - What went well? What
 - went wrong? What can
 - we improve?
 - These lessons are applied in the next sprint. This continuous
 - improvement cycle makes Agile powerful.
-

7. Deployment

- At regular intervals, working software increments are released to customers. The
 - product evolves sprint by sprint until the final version is ready.
-

Agile Activities (Core Practices)

Agile is not just about process—it involves a set of activities and practices:

1. Backlog Management → Maintaining and prioritizing features.
2. Sprint Planning → Selecting tasks for each sprint.

3. Daily Stand-ups → Team sync-ups to track progress.
 4. Incremental Development → Delivering small but working versions.
 5. Continuous Testing & Integration → Ensuring quality at all times.
 6. Customer Collaboration → Regular involvement for validation.
 7. Retrospectives → Continuous process improvement.
-

Analysis of Agile Process

Strengths / Advantages

1. Customer Satisfaction → Continuous delivery of working software keeps customers happy.
 2. Flexibility → Requirements can change at any stage.
 3. Reduced Risk → Problems are discovered early due to frequent feedback.
 4. High Quality → Continuous testing and integration maintain quality.
 5. Transparency → Customers and stakeholders always know progress.
 6. Motivated Teams → Agile encourages self-organizing and empowered teams.
 7. Faster Time-to-Market → Early releases give customers usable software quickly.
-

Weaknesses / Limitations

1. Not ideal for very large projects with unclear overall vision.
 2. Requires experienced developers and strong communication.
 3. Customer involvement is demanding → requires constant availability.
 4. Minimal documentation → future maintenance may be harder.
 5. Scope creep (uncontrolled changes) can occur due to flexible requirements.
 6. Less predictability in cost, time, and scope compared to Waterfall.
-

Agile Frameworks (Popular Methods)

- Scrum → Most widely used, sprint-based, roles like Scrum Master, Product Owner.
 - Kanban → Visual workflow management using boards and cards.
 - Extreme Programming (XP) → Focuses on engineering practices like TDD, pair programming.
 - Lean Software Development → Reduces waste and improves efficiency.
 - Crystal, DSDM, FDD → Other agile families.
-

Summary

- The Agile Development Process is iterative, incremental, and flexible.
- Work is divided into sprints/iterations.

- Key activities include planning, backlog management, design, coding, testing, review, and retrospectives.
- Agile promotes customer collaboration, rapid delivery, adaptability to change, and continuous improvement.
- It works best for dynamic, fast-changing, and customer-driven projects, but may not suit projects requiring strict documentation and fixed scope.

