

Introduction to Agile Software Development

Introduction to Agile Software Development

(A concise, easy-to-read reference guide)

1. What Is Agile?

Aspect	Description
Definition	A set of principles and practices for delivering high-quality software quickly and adaptively.
Goal	Respond to change, deliver value continuously, and foster collaboration among all stakeholders.
Origin	Formalized in the Agile Manifesto (2001) by 17 software practitioners. “Individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; responding to change over following a plan.”

2. Agile Values & Principles

Value	What It Means in Practice
Individuals & interactions	Empower self-organizing teams; encourage face-to-face communication.
Working software	Deliver usable increments every iteration (usually 1–4 weeks).
Customer collaboration	Involve customers/product owners continuously; adjust scope based on feedback.
Responding to change	Embrace new requirements, even late in development.

12 Principles (summary)

1. Satisfy the customer through early & continuous delivery.
2. Welcome changing requirements, even late.
3. Deliver working software frequently (weeks, not months).
4. Business people & developers must work together daily.
5. Build projects around motivated individuals; trust them.
6. Face-to-face conversation is the most efficient communication.

Introduction to Agile Software Development

7. Working software is the primary measure of progress.
 8. Agile processes promote sustainable development (constant pace).
 9. Continuous attention to technical excellence & good design.
 10. Simplicity—maximizing the amount of work not done—is essential.
 11. Best architectures, requirements, and designs emerge from self-organizing teams.
 12. Regularly reflect & adjust behavior to become more effective.
-

3. Common Agile Frameworks

Framework	Typical Iteration Length	Key Artefacts	Typical Use Cases
Scrum	2–4 weeks (Sprint)	Product Backlog, Sprint Backlog, Increment, Definition of Done	Teams needing clear roles (Product Owner, Scrum Master, Development Team).
Kanban	Continuous flow (no fixed timebox)	Kanban board, WIP limits, Cumulative Flow Diagram	Maintenance, support, or teams that prefer visual workflow control.
XP (Extreme Programming)	1–2 weeks	User Stories, Pair Programming, Test-Driven Development (TDD), Continuous Integration	Projects demanding high technical discipline & frequent releases.
Lean Software Development	Flow-oriented	Value Stream Mapping, Eliminate Waste, Pull System	Organizations focused on efficiency and waste reduction.
SAFe (Scaled Agile Framework)	Program Increment (8–12 weeks)	Agile Release Train, PI Planning, Portfolio Backlog	Large enterprises needing coordination across many Agile teams.

You can mix elements (e.g., Scrum + XP technical practices) to suit your context.

4. Core Roles & Responsibilities

Introduction to Agile Software Development

Role	Primary Responsibilities
Product Owner	Owns the Product Backlog , prioritizes items based on business value, clarifies requirements, accepts/rejects increments.
Scrum Master / Agile Coach	Facilitates the process, removes impediments, coaches the team on Agile practices, protects the team from external disruptions.
Development Team	Cross-functional members who design