1. What is Agile? (Summary of All Documents)

Agile is a **flexible, iterative approach** to project management that focuses on:

- Delivering **value early & frequently**
- **Continuous customer involvement**
- Welcoming **changes at any stage**
- Breaking big projects into **small iterations (sprints)**
- Teams that are **self-organizing & collaborative**

Agile was created in **2001** by 17 software engineers who published the **Agile Manifesto**.
Its goal: **Respond to change quickly in a fast-moving, technology-driven world.**

Example:
You want to build a mobile app with 20 features.
Instead of planning all 20 features for 6 months (Waterfall), Agile builds 2–3 features in 2 weeks, gets user feedback, then improves.

---

2. 4 Values of the Agile Manifesto

(Extracted from your files)

subtitle (3)

Value 1: Individuals & Interactions > Processes & Tools

People working together matter more than rigid tools.

Example:
Instead of long email chains, the team has a short 10-minute stand-up meeting every morning.

Value 2: Working Software > Comprehensive Documentation

Focus on **building usable output**, not drowning in documents.

Example:
Instead of a 50-page report, developers deliver a working login page in week 1.

Value 3: Customer Collaboration > Contract Negotiation

Customers stay involved **daily**, not only at the beginning.

Example:
The customer reviews demo every 2 weeks and changes priorities anytime.

Value 4: Responding to Change > Following a Plan

Agile welcomes change—even late in the project.

Example:
If users suddenly prefer a dark mode UI, Agile adjusts immediately.

---

3. The 12 Agile Principles (Simplified + Examples)

(Extracted fully from the document)

subtitle (4)

---

1. Deliver value early & continuously

Example: Release a basic app in 2 weeks so customers can start using it early.

---

2. Welcome changing requirements—even late

Example: Halfway through development, a new rule comes—Agile adapts.

---

3. Deliver working software frequently

Example: Every 2 weeks, release a small usable feature.

---

4. Business & developers must work together daily

Example: Daily stand-ups ensure no miscommunication.

---

5. Build projects around motivated people; trust them

Example: Give the team autonomy—they choose how to complete tasks.

---

6. Face-to-face communication is best

Example: Quick meeting > 20 Slack messages.

---

7. Working product is the measure of progress

Example: Feature delivered > documentation updated.

---

8. Support sustainable development (no burnout)

Example: Instead of 14-hour shifts, maintain a steady pace of 8-hour work.

---

9. Focus on technical excellence & good design

Example: Clean code now prevents rework later.

---

10. Simplicity—do only what is necessary

Example: Build MVP first; don't add complex features before needed.

---

11. Self-organizing teams create the best solutions

Example: Team decides roles (who tests, who codes) on their own.

---

12. Regularly reflect, then improve (retrospective)

Example: After every sprint, team discusses what went well/poorly.

---

4. AGILE FRAMEWORKS

Your documents mention the four major Agile methods: **Scrum, Kanban, Lean, XP**.
Here is a complete, clean summary with examples.

A. SCRUM

Scrum is the **most popular Agile framework**, based on 2-week cycles called **sprints**.
(Extracted from your files)

subtitle (5)

Scrum Roles

1. **Product Owner** – Prioritizes work (product backlog)
2. **Scrum Master** – Removes blockers; servant leader
3. **Development Team** – Builds the product

Scrum Events

- **Sprint Planning** – Choose tasks for the sprint
- **Daily Stand-up** – 10-minute sync meeting
- **Sprint Review** – Show completed work
- **Sprint Retrospective** – Improve for next sprint

Scrum Example

Building an e-commerce app:

Sprint 1 → Login + Signup
Sprint 2 → Product listing
Sprint 3 → Cart system
Sprint 4 → Payment integration

Every 2 weeks, customer gives feedback → improvement happens instantly.

B. KANBAN

Kanban uses a **visual board** to track work in real time.
(Extracted from your files)

subtitle (5)

Kanban Columns

- Backlog

- To Do
- In Progress
- Review
- Done

Kanban Benefits

✓ Great for continuous work
✓ Easy to see bottlenecks
✓ No fixed sprint

Kanban Example

Writing a research paper:

Backlog	**In Progress**	**Done**
Literature review	Writing Introduction	Summary finished

Work moves across the board like Post-it notes.

C. LEAN (Origin of Agile Thinking)

Lean focuses heavily on **removing waste** and increasing **efficiency**.
(Extracted from your file: Agile frameworks are derived from Lean)

subtitle (6)

Types of Waste Lean Eliminates

- Waiting time
- Unnecessary tasks
- Over-processing
- Too much documentation
- Rework

Lean Example

Instead of lengthy approval chains, the team gets direct sign-off from the manager.

D. XP (EXTREME PROGRAMMING)

XP focuses on **high-quality coding** and fast feedback cycles.
(Extracted from your files)

subtitle (5)

Key XP Practices

- **Pair Programming** (two developers code together)
- **Test-Driven Development (TDD)**
- **Continuous Integration**
- **Refactoring** (improving code continuously)

XP Example

Two programmers sit together:

One writes code → the other reviews instantly → fewer bugs, higher quality.

---

5. How All Frameworks Relate to Lean

Your text says all Agile frameworks are derived from Lean.
✓ Kanban is from Lean manufacturing
✓ Scrum and XP use Lean principles
✓ All focus on removing waste and delivering quickly

(Reference)

subtitle (6)

---

6. Agile vs Predictive (Waterfall) — Key Differences

Agile	**Predictive/Waterfall**
Iterative	Sequential
Welcomes change	Avoids change

Agile	Predictive/Waterfall
Customer involved daily	Customer only at start/end
Delivers small pieces frequently	Delivers whole product at end
Flexible scope	Fixed scope
Fast feedback	Slow feedback

---

7. Complete Final Summary (Short + Powerful)

Agile is a flexible framework designed to deliver **value early, often, and with customer involvement**.
It operates through short cycles (sprints), prioritizes people, and welcomes change.
Scrum handles iterative development, Kanban visualizes workflow, Lean eliminates waste, and XP ensures technical excellence.

Agile is governed by **4 values** and **12 principles** that emphasize collaboration, simplicity, sustainability, and continuous improvement.