Discover SCRUM

The Agile Framework That Conquered the Tech World

LearnWithUs

Why Agile?

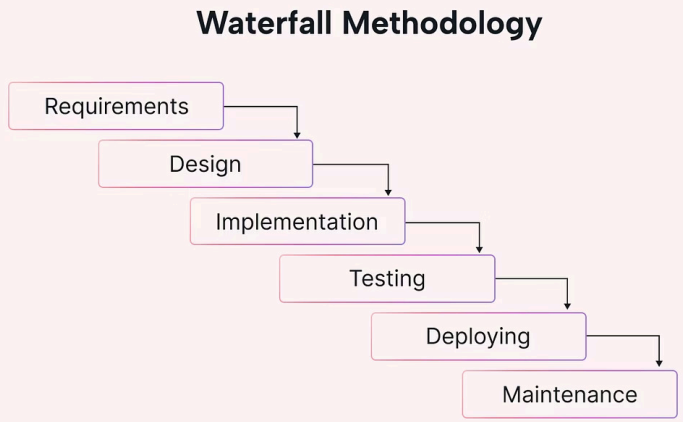
The Problem with the V-Model

Traditional project management followed a rigid, sequential process: define everything upfront, build it, then test it — often months later. By the time the product was delivered, requirements had changed, budgets were blown, and customers were disappointed.

The Agile Manifesto (2001)

In February 2001, 17 software pioneers gathered in Snowbird, Utah, and signed the **Agile Manifesto** — a declaration of a better way to build software. Their goal: shorter cycles, tighter feedback loops, and teams empowered to adapt quickly to change.

> The manifesto defined 4 core values and 12 principles that still guide modern software teams today.



The 4 Values of the Agile Manifesto

The manifesto doesn't reject processes or tools — it simply prioritizes what matters most when the two sides conflict.

Individuals & Interactions

Over processes and tools

Working Software

Over comprehensive documentation

Customer Collaboration

Over contract negotiation

Responding to Change

Over following a plan

The items on the right still have value — but the items on the left are valued **more**.

What is SCRUM?

SCRUM is a lightweight **agile framework** designed to help teams deliver valuable products through short, focused cycles called **Sprints**. It provides a simple structure of roles, rules, and rituals that enable teams to self-organize, inspect their progress, and continuously improve.

Originally developed by Ken Schwaber and Jeff Sutherland in the early 1990s, SCRUM is today the **most widely adopted agile framework** in the world — used by startups and Fortune 500 companies alike, well beyond software development.

87%

of agile teams

use SCRUM or a SCRUM-hybrid approach

2-4

week Sprints

typical cycle length for most teams

SCRUM in One Sentence

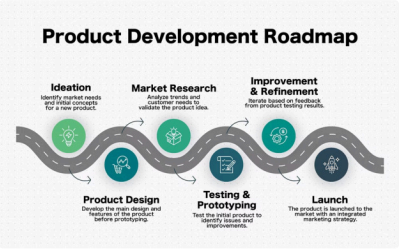
> A team of cross-functional people builds a product increment every Sprint, inspects results, and adapts their plan — repeat until done.

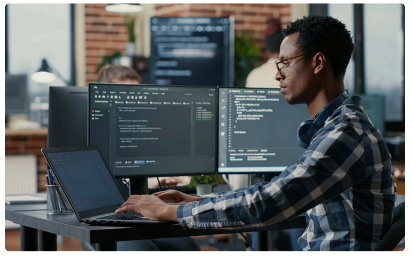
Why SCRUM Wins

- Simple rules, powerful outcomes
- Fast feedback from real users
- Built-in risk reduction
- Scales from 3 to 3,000 people

The 3 SCRUM Roles

Every SCRUM team has exactly three roles. Each has a distinct focus — together they cover the full lifecycle of product delivery.





Product Owner

The **voice of the customer**. The PO owns the Product Backlog, prioritizes features by business value, and ensures the team always builds the right thing. They make decisions so the team never waits.

Scrum Master

The **servant-leader** of the team. The SM removes blockers, coaches the team on SCRUM practices, and protects the team from external distractions. They facilitate every ceremony but don't manage the team.

Development Team

The **builders**. A self-organizing, cross-functional group of 3–9 people who design, build, and test the product. Everyone on the team is a "developer" regardless of their specialty — no siloes allowed.

The 3 SCRUM Artifacts

Artifacts are the key **documents and deliverables** that give the entire team visibility into the work. Each artifact has a specific commitment that defines what "done" looks like.

1	2	3
Product Backlog The **master list** of everything the product might ever need — features, fixes, improvements, and experiments. It is owned by the Product Owner, always prioritized, and never truly "finished." Items at the top are refined and ready; items at the bottom are rough ideas.	**Sprint Backlog** The **team's commitment** for a single Sprint. It contains the items selected from the Product Backlog plus a plan for delivering the Sprint Goal. It is updated daily and belongs entirely to the Development Team — no one can change it mid-Sprint without the team's agreement.	**Increment** The **tangible output** of each Sprint — a working, tested, potentially shippable piece of the product. Each Increment must meet the team's Definition of Done. Increments stack on top of each other, growing the product with every Sprint.

The 5 SCRUM Ceremonies

Ceremonies are structured meetings that create rhythm, alignment, and continuous improvement. Each has a fixed time-box to keep the team focused.

🗓️ Sprint Planning

Up to 8 hours (2-week Sprint). The team selects backlog items and defines the Sprint Goal together.

☀️ Daily Stand-up

15 minutes, every day. Three questions: What did I do? What will I do? Any blockers?

🔍 Sprint Review

Up to 4 hours. The team demos the Increment to stakeholders and collects feedback.

🔄 Retrospective

Up to 3 hours. The team reflects: What went well? What to improve? One action item minimum.

✏️ Backlog Refinement

Up to 2 hours. The team grooms upcoming backlog items — clarifying, estimating, and splitting stories.

A Typical 2-Week Sprint

Every Sprint follows the same heartbeat — a predictable cycle that gives teams stability while delivering continuous value to the business.

Sprint Planning

Day 1 — up to 8h

Backlog Refinement

Mid-sprint — up to 2h

Daily Stand-ups

Days 2–9 — 15 min daily

Review & Retro

Day 10 — review then retro

Key Principle: The Sprint is a **time-box** — its end date never moves. Scope can be reduced, but the Sprint always ends on time. This creates a forcing function for delivery.

Sprint Goal: Every Sprint has a single, clearly stated goal that gives the team focus. Individual stories may change; the goal does not.

User Stories & Estimation

Writing a User Story

A User Story captures a feature from the **end user's perspective**, not a technical specification. The standard format ensures every story has a who, a what, and a why.

> **As a** registered user,
> **I want** to reset my password by email,
> **so that** I can regain access to my account if I forget it.

Good stories are **INVEST**: Independent, Negotiable, Valuable, Estimable, Small, and Testable. They are enriched with **Acceptance Criteria** — specific conditions that must be true for the story to be "done."

Planning Poker 🃏

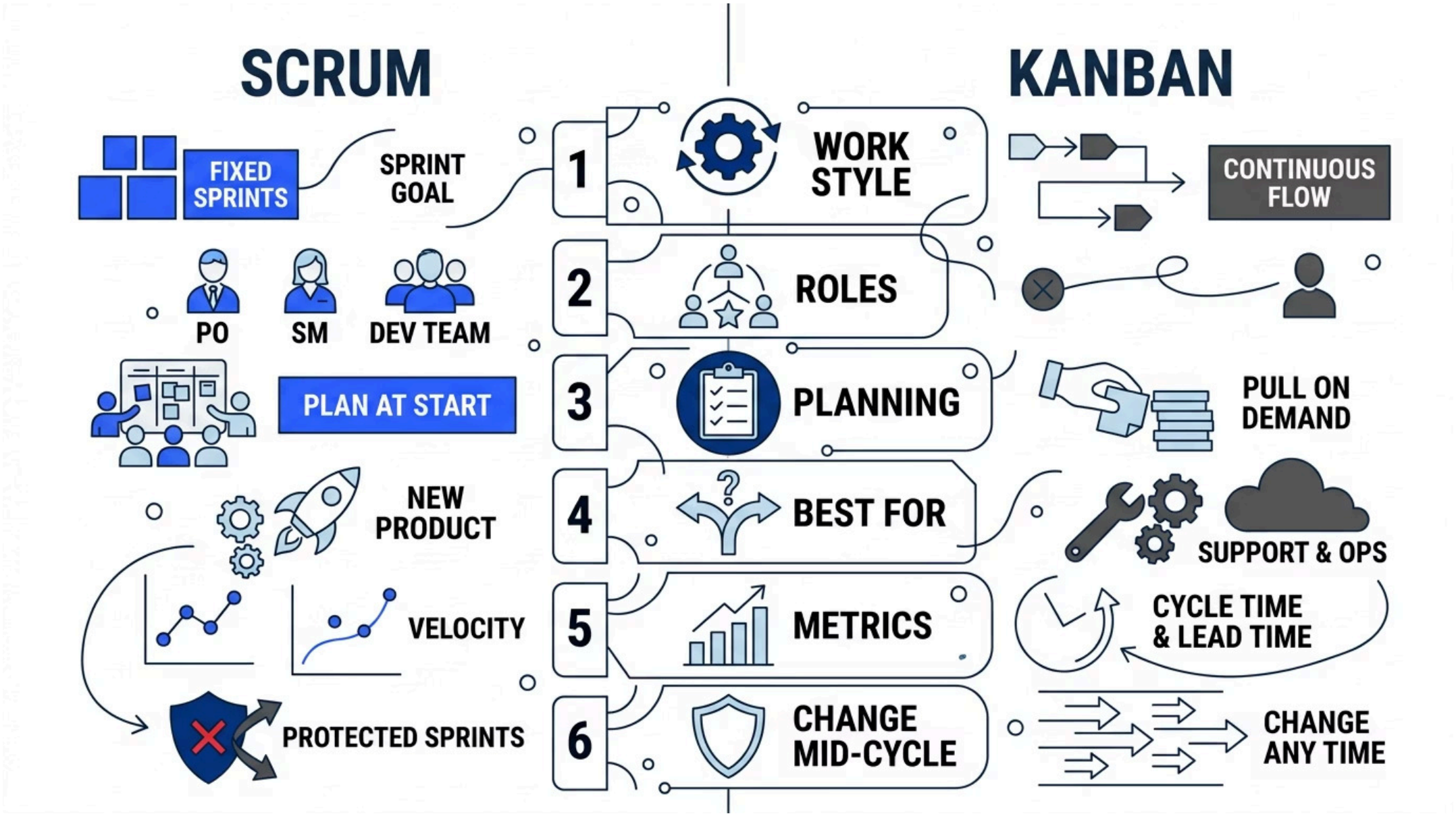
Planning Poker is a consensus-based estimation technique. Each team member holds a set of cards with values based on the **Fibonacci sequence** (1, 2, 3, 5, 8, 13, 20…). Everyone reveals their card simultaneously to avoid anchoring bias.

- Large gaps trigger discussion — why the difference?
- The team re-estimates until consensus is reached
- Estimates are in **Story Points**, not hours
- Story Points measure complexity, not time

Story Points are relative — a 5-point story is roughly twice as complex as a 2-point story, regardless of how many hours it takes.

SCRUM vs. Kanban

Both are popular agile frameworks — but they serve different contexts. Choosing the right one depends on the nature of your work and team structure.



Choose SCRUM when you are building a product with a dedicated team, evolving features, and clear Sprint Goals every 2 weeks.

Choose Kanban when your team handles ongoing support, bug fixes, or operational work where new tasks arrive unpredictably every day.

Many mature teams use a hybrid — "Scrumban" — combining Sprint cadence with a Kanban board for day-to-day task flow.

Common SCRUM Pitfalls

Avoid these 3 common mistakes to maximize the benefits of your SCRUM implementation.

The "ScrumBut"

Adopting SCRUM without fully practicing it. Ignoring fundamental principles or skipping ceremonies weakens effectiveness. The "but" after "we do Scrum" is a warning sign.

The "Scrum Zombie"

Following the SCRUM rituals (meetings, artifacts) without embracing the Agile spirit. Teams go through the motions but lack self-organization, transparency, or continuous improvement.

The Feature Factory

Prioritizing the rapid delivery of features over their real value. Building without validation or user feedback leads to creating the wrong product. "More is better" is not always true.

Master SCRUM with LearnWithUs Premium

Ready to transform your SCRUM expertise into real competitive advantage? LearnWithUs Premium gives you the tools and knowledge to excel.

Advanced SCRUM Frameworks

Dive deep into Scaled Agile (SAFe), LeSS, Nexus, and enterprise SCRUM implementations. Master complex scaling patterns and best practices for large organizations.

Real-World Projects & Simulations

Put your skills to the test on real-world problems through our hands-on training sessions and guided projects. Put theory into practice.

Exclusive Access

Take advantage of monthly webinars led by experts, early access to new features, and priority support for all your questions.

MODULE COMPLETED ✓

LearnWithUs