



# The Agile Manifesto: Four Core Values That Transformed Software Development

In 2001, seventeen visionary developers gathered in Snowbird, Utah, to revolutionize how we build software. Their breakthrough: the Agile Manifesto, a set of values and principles that prioritizes people, collaboration, and adaptability over rigid processes and documentation.



**by Kimberly Wiethoff, MBA, PMP, PMI-ACP**

[Managing Projects The Agile Way](#)

#AgileManifesto #AgileValues #IndividualsOverProcesses #WorkingSoftware  
#CustomerCollaboration #RespondToChange #AgilePrinciples #AgileDevelopment #AgileMindset  
#AgileTransformation #ManagingProjectsTheAgileWay



# The Birth of a Movement

## The Problem

Traditional software development was failing teams and customers alike. Rigid methodologies, excessive documentation, and inflexible plans created frustration and delayed delivery of value.

Developers across various methodologies recognized a common thread: the need for a more human-centered, responsive approach to building software.

## The Solution

In February 2001, seventeen developers came together seeking common ground. What emerged was not another methodology, but a set of values that would reshape the entire industry.

The Agile Manifesto represented a fundamental shift in thinking—from process-heavy bureaucracy to people-focused collaboration.

# The Four Core Values

At its heart, the Agile Manifesto presents four powerful statements that challenge conventional wisdom. These values don't reject traditional practices—they simply prioritize what matters most.

## Individuals and Interactions

over processes and tools

## Working Software

over comprehensive documentation

## Customer Collaboration

over contract negotiation

## Responding to Change

over following a plan



# Value #1: Individuals and Interactions Over Processes and Tools

People are at the absolute heart of software development. While processes and tools certainly have their place, they cannot replace the power of human communication, creativity, and collaboration.

Agile recognizes that software engineers create solutions for real people—users who want to improve their lives and achieve their goals. Without genuine communication and collaboration between developers and users, even the most sophisticated tools and processes would fail to deliver what's truly needed.

This value emphasizes fostering a collaborative and empowered team environment where individuals are valued for their unique contributions. When team members interact freely and openly, innovation flourishes and problem-solving becomes more effective.



# Why People Matter More Than Process



## Innovation Through Interaction

Breakthrough solutions emerge from conversations, not from following rigid procedures.



## Empowered Teams

When individuals feel valued and heard, they take ownership and drive results.



## Direct Communication

Face-to-face conversations solve problems faster than any documentation or tool.



## Continuous Learning

Teams that interact openly share knowledge and grow together organically.

## Value #2: Working Software Over Comprehensive Documentation

While documentation certainly has its place and provides value, Agile prioritizes delivering functional, working software that customers can actually use. This doesn't mean abandoning documentation entirely—it means being smart about what we document and when.

The Agile approach provides just enough documentation to begin developing the final product. As the software evolves through development, each delivered increment is evaluated, and additional documentation is created only as needed. This streamlined approach avoids the lengthy delays that plagued traditional Waterfall methodology.

Instead of months spent creating technical specifications, requirements documents, design documents, and extensive test plans before writing a single line of code, Agile teams focus on delivering working software that demonstrates real value to customers early and often.

# The Documentation Dilemma

1

## Traditional Approach

- Extensive upfront documentation
- Months of planning before coding
- Documents often outdated by delivery
- Delayed feedback from users

2

## Agile Approach

- Just-enough documentation
- Working software in weeks
- Living documentation that evolves
- Immediate user feedback





# Value #3: Customer Collaboration Over Contract Negotiation



In the early days of software development, contracts were king. Detailed product specifications would be negotiated and locked into contracts, creating a rigid framework that often led to conflict.

The problem? Most of the time, the end result created a disconnect between what the contract stated, what the product actually did, and what the customer truly needed. Requirements evolved, priorities shifted, but contracts remained static.

Agile transformed this dynamic by emphasizing continuous development with direct customer feedback loops. The product roadmap begins with a high-level concept and progresses through an iterative process, incorporating customer input at every stage.



# The Power of Continuous Collaboration



This iterative cycle ensures the product evolves to meet customer expectations, not just contractual obligations. Collaboration becomes the foundation of success, replacing adversarial contract negotiations with partnership.

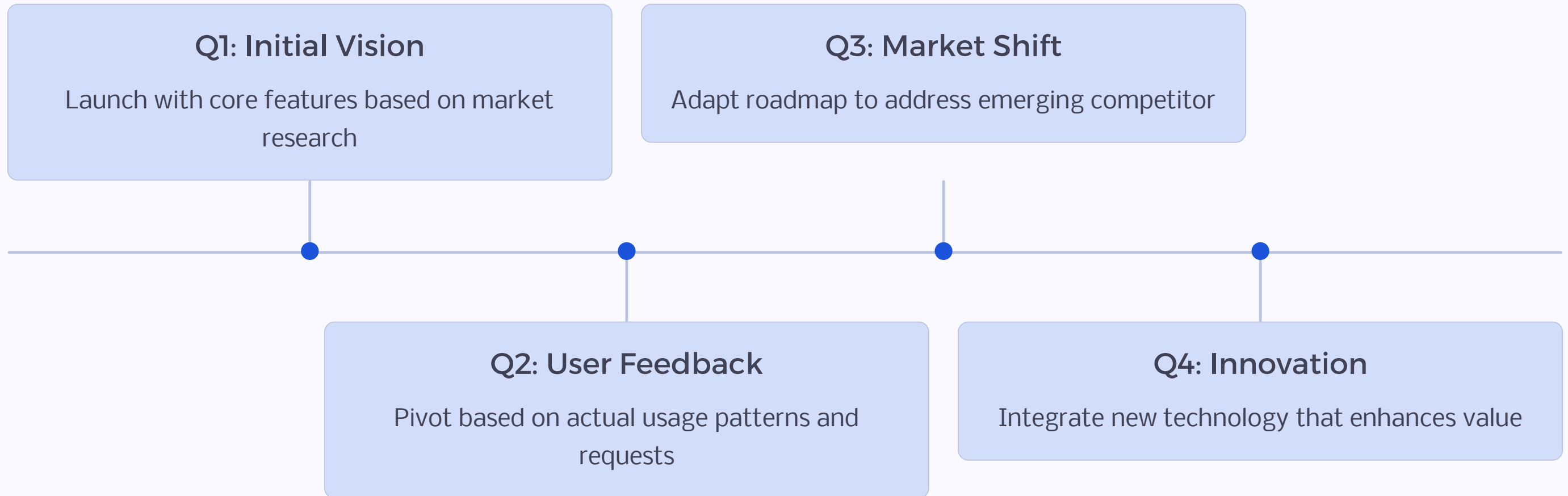
## Value #4: Responding to Change Over Following a Plan

Traditional software development relied on static plans and roadmaps designed and documented upfront. Development efforts were expected to adhere rigidly to these plans, regardless of changing circumstances. This inflexible approach proved inefficient in a world where software needs, requirements, and priorities constantly evolve.

The Agile Manifesto recognizes that change is not a disruption—it's an opportunity. Software development teams should have the ability to change direction and pivot when requirements shift or new information emerges.

With dynamic roadmaps that can be updated monthly or quarterly, Agile teams adapt fluidly to changing conditions. This flexibility allows organizations to respond to market shifts, customer feedback, and technological advances without being constrained by outdated plans.

# Embracing Change as Opportunity



- 📌 **Key Insight:** In Agile, change is viewed as an opportunity for learning and improvement, not as a setback or failure. Teams that embrace this mindset can navigate uncertainty with confidence.

# The Twelve Principles: Practical Guidance

Beyond the four core values, the Agile Manifesto includes twelve principles that provide actionable guidance for implementation. These principles translate philosophy into practice.

<b>Deliver Value Early and Often</b> Satisfy customers through continuous delivery of valuable software	<b>Welcome Changing Requirements</b> Even late in development, harness change for competitive advantage
<b>Deliver Frequently</b> From weeks to months, with preference for shorter timescales	<b>Work Together Daily</b> Business people and developers must collaborate throughout the project
<b>Support Motivated Individuals</b> Build projects around motivated people in trusted environments	<b>Face-to-Face Conversation</b> The most efficient method of conveying information to and within teams



# The Twelve Principles: Continued

## **Working software is the primary measure of progress**

Rather than focusing on documentation or process compliance, Agile teams are measured by their ability to deliver functional, valuable software.

## **Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely**

Sustainable pace prevents burnout and enhances long-term productivity. Teams should deliver value consistently without excessive overtime or stress.

## **Continuous attention to technical excellence and good design enhances agility**

Clean code, refactoring, and well-architected systems allow teams to adapt quickly to change. Technical debt slows progress—Agile keeps it in check.

## **Simplicity—the art of maximizing the amount of work not done—is essential**

Agile values simplicity and focus. Eliminating non-essential work allows teams to concentrate on what truly delivers value to the customer.

## **The best architectures, requirements, and designs emerge from self-organizing teams**

Agile trusts teams to make the right decisions. When team members are empowered to collaborate and innovate, the result is often better than top-down directives.

## **At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly**

Retrospectives and continuous improvement are central to Agile. Teams pause regularly to evaluate what's working and what isn't—and make changes to improve performance.

# The Lasting Impact of Agility



Since its inception in 2001, the Agile Manifesto has catalyzed a paradigm shift in how software is developed worldwide. It has empowered teams to embrace uncertainty, welcome change, and deliver value with unprecedented speed and efficiency.

Agile methodologies like Scrum, Kanban, and Extreme Programming have become ubiquitous in the tech industry, fundamentally reshaping organizational structures, processes, and cultures. What started with seventeen developers has influenced millions.

## 20+

Years of Impact

Transforming software development since  
2001

## 71%

Organizations Using Agile

Majority of companies have adopted Agile  
practices

## 3x

Faster Delivery

Agile teams typically deliver value three  
times faster


# Embrace the Agile Mindset


"We are uncovering better ways of developing software by doing it and helping others do it."


As we reflect on the Agile Manifesto and its enduring legacy, we're reminded of the transformative power of agility in driving innovation, collaboration, and customer satisfaction. By embracing these values and principles, organizations can navigate the complexities of the digital age with confidence and resilience.

The Agile Manifesto isn't just about software development—it's about creating better ways of working together, delivering value, and adapting to change. Whether you're a developer, product manager, or business leader, these principles offer a path toward more effective, human-centered work.

 **Start with people—invest in communication and collaboration**

 **Build partnerships with customers through continuous feedback**

 **Focus on delivering working solutions, not perfect documentation**

 **Welcome change as an opportunity for improvement and learning**