

# Agile Values

## What are the 4 Agile Values?

Agile Values refers to the set of 4 values outlined by the Agile Alliance in [The Agile Manifesto](#). This set of values encourages putting people before processes, getting software out the door fast, collaborating with customers, and adjusting plans as needed.

- 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 plan

Below we will break down each of the 4 Agile Values and explain how they may be interpreted through the lens of a product manager or [product owner](#).

## Individuals and Interactions Over Processes and Tools

No matter how well-researched your process and high-tech your tools are, it's the team you work with and the way you work together that determines success. Your team and their ability to communicate effectively and efficiently is more valuable than the processes they follow or the tools you use.

This isn't to say that agile philosophies discourage formalized processes or tools. Both can be helpful in providing structure for your team and facilitating interactions. But at the end of the day, they come second.

After all, processes and tools are worthless if your team can't communicate. But put a smart, motivated team up to a task without any processes or tools to manage the project and chances are they'll find some way to get it done.

## Working Software Over Comprehensive Documentation

Traditional product development processes often required extensive documentation before a single line of code was written. Under the agile philosophy, [getting software in the hands of customers](#) is the highest priority. After all, how are you going to improve your product if you don't get it out in the wild and collect feedback from real users?

While this value highlights the importance of shipping software over letting documentation be a bottleneck, it's important to note that documentation in itself is not a bad thing...as long as you don't overdo it.

## Customer Collaboration Over Contract Negotiation

The agile philosophy highlights the importance of [customer-centric product development practices](#) over product-centric approaches. While contracts will always have their place in business, a list of the things you're offering your customer is no replacement for actually communicating with them about what their needs are and where their challenges are.

Traditional product-centric processes allowed contracts to dictate what was delivered in the end, which left a lot of room for mismatched expectations. The agile philosophy (and many of the formalized processes that have come out of it) encourage building a [continuous customer feedback loop](#) into development cycles.

Under the agile philosophy, customer collaboration begins early in the development process and happens frequently throughout. This culture of close collaboration with real customers helps product people ensure they're delivering effective, useful solutions to customers. When you talk to customers often and build feedback into your development process, you reduce risk and eliminate guesswork.

## Responding to Change Over Following a Plan



改组，革新

An important benefit of the agile methodology is that it encourages [frequent reviewing and retooling of current plans](#) based on new information that the team is continually gathering and analyzing. The product roadmap, then, is no longer a static document, but a [dynamic](#) strategy. Product managers in agile environments will need to learn to present their dynamic roadmaps to stakeholders in a transparent manner that reflects the likelihood of change based on new learnings.

In other words, the agile methodology lets a product team adjust its priorities and plans whenever doing so makes strategic sense. These teams do not get stuck in an outdated plan simply because they have committed to seeing it through.

## Remember: Agile Values are not Rules

Finally, don't forget that [agile](#) is a mindset, not a set of strict rules to follow. These values are up for interpretation and are not solid instructions set in stone.

Keep learning about the agile values in our webinar, Managing Complex Requirements in an Agile World, below.



# Subscribe to our Newsletter

Your email address

Subscribe

ProductPlan is committed to protecting and respecting your privacy, and we'll only use your personal information to administer your account and to provide the products and services you requested from us. From time to time, we would like to contact you about our products and services, as well as other content that may be of interest to you. If you consent to us contacting you for this purpose, please tick below to say how you would like us to contact you:

☐ I agree to receive other communications from ProductPlan.

In order to provide you the content requested, we need to store and process your personal data. If you consent to us storing your personal data for this purpose, please tick the checkbox below.

☐ I agree to allow ProductPlan to store and process my personal data.

You may unsubscribe from these communications at any time. For more information on how to unsubscribe, our privacy practices, and how we are committed to protecting and respecting your privacy, please review our [Privacy Policy](#).

### PRODUCT

- Features
- Pricing
- Enterprise
- Integrations
- Templates
- Security
- Release Notes
- Status

### COMPANY

- About Us
- Contact Us
- Careers
- Privacy Policy
- Terms of Service
- API Docs

### RESOURCES

- Learning Center
- Blog
- Glossary
- Downloads
- Webinars
- Support

### ULTIMATE GUIDES

- Product Roadmaps
- Product Managers
- Product Management
- Product Planning
- Product Management Frameworks
- Product Strategy
- Resources for Product Managers
- Product Launch