Agile Values and Principles

The four values of the Agile Manifesto are:

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

What are the Agile Principles?

There are 12 Agile principles outlined in The agile Manifesto in addition to the 4 agile values. These 12 principles for agile software development help establish the tenets of the agile mindset. They are not a set of rules for practicing agile, but a handful of principles to help instill agile thinking.

## Agile Principle 1

### **“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”**

The best ways to ensure you make customers happy while continuously delivering valuable software are to ship early, iterate frequently, and listen to your market continually.

Unlike traditional approaches to product development, which have notoriously long development cycles, agile principles encourage minimizing the time between ideation and launch. The idea is to get a working product in the hands of customers as soon as possible. Doing this successfully means product managers are able to quickly get a [good](https://www.productplan.com/learn/minimum-viable-product/) product out and into the world and use it to get feedback from real customers. This feedback is then fed back into the product development process and used to inform future releases.

## Agile Principle 2

### **“Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.”**

In the world around us, change is the only constant. Agile principles and values support responding to these changes rather than moving forward in spite of them. Previous approaches to product development were often change adverse; detailed, well-documented plans were made before development began and were set in stone regardless of new findings. Agile principles support observing changing markets, customer needs, and competitive threats and changing course when necessary.

**How it looks in practice:**

* Product teams are guided by high-level strategic goals and perhaps even [themes](https://www.productplan.com/learn/organize-your-roadmap-by-themes/) below those goals. The product department’s success is measured against progress toward those strategic goals rather than by delivery of a predefined feature set.
* Product constantly has its ear to the ground monitoring the market, customer feedback, and other factors which could influence product direction. When actionable insight is uncovered, plans are adjusted to better serve customer and business needs.
* Product strategy and tactical plans are reviewed, adjusted, and shared on a regular cadence to reflect changes and new findings. As such, product needs to manage the expectations of executive stakeholders appropriately and ensure they understand the why behind changes.

## Agile Principle 3

### **“Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”**

Agile philosophy favors breaking a product’s development into smaller components and “shipping” those components frequently. Using an agile approach, therefore — and building in more frequent mini-releases of your product— can speed the product’s overall development.

This agile approach, with short-term development cycles of smaller portions of the product, results in less time spent drafting and poring over the large amounts of documentation that characterizes Waterfall product development. More importantly, this frequent-release approach creates more opportunities for you and your teams to validate your product ideas and strategies from the qualified constituencies who see each new release.

**How it looks in practice:**

* Agile development cycles, often called “sprints” or “iterations” break down product initiatives into smaller chunks that can be completed in a set timeframe. Often this timeframe is between 2 and 4 weeks which truly is a sprint if you consider the marathon-like development cycles waterfall teams often follow.
* Another popular alternative to agile sprints is continuous deployment. This method of shipping software frequently works less in terms of predetermined time boxes and more in terms of simply deciding what to do and doing it.

## Agile Principle 4

### **“Business people and developers must work together daily throughout the project.”**

Communication is a critical component of any project or team’s success, and agile principles essentially mandate that it’s a daily event. It takes a village to raise a child they say, and that applies to product as well.

A successful product requires insight from the business and technical sides of an organization which can only happen if these two teams work together consistently. Regular communication between business people and developers helps improve alignment across the organization by building trust and transparency.

**How it looks in practice:**

* Cross-functional agile product development teams include product people. This means that product is represented on the development team and bridges the gap between technical and business aspects of the product.
* Daily update meetings, or standups, are one technique many agile shops use to put this principle in practice and keep everyone connected.

## Agile Principle 5

### **“Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”**

A key part of the agile philosophy is empowering individuals and teams through trust and autonomy. The agile team needs to be carefully built to include the right people and skill sets to get the job done, and responsibilities need to be clearly defined before the beginning of a project. Once the work has begun, however, there’s no place in agile for micromanagement or hand holding.

**How it looks in practice:**

* Product must clearly ensure engineering understands strategy and requirements before development starts. This means not only sharing user stories with the cross-functional team but also the bigger picture outlined in the product roadmap.
* Product is not responsible for explaining “how” something should be built. They need to share what and why, but it’s the delivery team’s job to determine the how. Furthermore, during sprints product does not micromanage outcome, instead they make themselves available to answer questions and provide support as needed.

## Agile Principle 6

### **“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”**

With so many distributed or [remote development teams](https://www.productplan.com/learn/remote-product-teams/)these days, this principle gets a bit of critique. But at the root of it, effective communication with developers means getting these conversations out of Slack and email and favoring more human interaction (even if done by video conference calls). The overall objective behind this principle is to encourage product people and developers to truly communicate in real time about the product, requirements, and the high-level strategy driving those things.

**How it looks in practice:**

* Daily standup meetings
* Collaborative [backlog grooming sessions](https://www.productplan.com/glossary/backlog-grooming/)
* Sprint planning meetings
* Frequent demos
* Pair-programming

## Agile Principle 7

### **“Working software is the primary measure of progress.”**

Proponents of the agile philosophy are quick to remind us that we’re in the business of building software, and that’s where our time should be spent. Perfect, detailed documentation is secondary to working software. This mentality pushes to get products to the market quickly rather than let documentation or an “it’s not done until it’s perfect” mentality become a bottleneck. The ultimate measure for success is a working product that customers love.

**How it looks in practice:**

* Designing and releasing “Minimum Viable Features” rather than fully-developed feature sets means thinking first and foremost about the smallest things we can ship to start getting customer feedback and validate as we continue to build software.
* A fail fast mentality means moving forward even in times of uncertainty and testing ideas rapidly.
* Ship software often: a useful product now is better than a perfect one later.

## Agile Principle 8

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

Keeping up with a demanding, rapid release schedule can be taxing on a team. Especially if expectations are set too high. Agile principles encourage us to be mindful of this and set realistic, clear expectations. The idea is to keep morale high and improve work-life balance to prevent burnout and turnover among members of cross functional teams.

**How it looks in practice:**

* Before every sprint, careful consideration of the amount of work that can be committed to is made. Development teams don’t over promise on what they can and cannot deliver. Effort estimations are a common practice in setting output expectations for development teams.
* Everyone agrees on what will get done during a sprint. Once a sprint has begun, no additional tasks are to be added except in rare cases.
* Product managers should act as gatekeepers to reduce the noise from other stakeholders and to avoid squeezing in additional unplanned work during an ongoing sprint.
* Product people should do their part in promoting a sense of psychological safety across the cross-functional team that encourages open communication and freely flowing feedback.

## Agile Principle 9

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

While the agile philosophy encourages shorter cycles and frequent releases, it also puts emphasis on the importance of keeping things neat and tidy so they don’t cause problems in the future. Product managers often forget about this aspect of development because they mostly don’t spend their days wading through their products’ codebases, but it is still of the utmost importance to them.

**How it looks in practice:**

* The team needs to be cognizant of [technical debt](https://www.productplan.com/glossary/technical-debt/) and the technical debt implications of any new features or initiatives added to the backlog. Developers and product need to work together to understand if and when technical debt is acceptable.
* On a regular basis, product will need to allocate development resources to refactoring efforts. Refactoring cannot be an afterthought, it needs to be an ongoing consideration.

## Agile Principle 10

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

You’ve probably heard of the 80/20 rule—the concept that you can usually get 80% of your intended results with just 20% of the work. Agile principles encourage thinking this way; doing the things that can have the most impact. In a product management context this means having a laser sharp focus on organizational objectives and making some cutthroat [prioritization decisions](https://www.productplan.com/learn/product-management-frameworks/). Agile principles discourage building merely for the sake of building by emphasizing the importance of being strategic and building with purpose.

**How it looks in practice:**

* Product managers need to make very focused product decisions and closely align product strategy with organizational goals while being extremely picky about what user stories and features actually make the cut. Using prioritization techniques to prioritize initiatives by effort and predicted impact is one way product teams can apply this agile principle to product development.
* The short sprints that agile is characterized by present many opportunities for rapid testing and experimentation which can help reduce uncertainty around whether initiatives will truly have the predicted impact. Using experiments to validate ideas before building them up to spec is a great way to weed out bad ideas and identify good ones.

## Agile Principle 11

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

In traditional software development methodologies, you’ll often see pyramid shaped teams where management makes key decisions for contributors. Agile principles suggest the use of self-organizing teams which work with a more “flat” management style where decisions are made as a group rather than by a singular manager or management team. The concept ties into agile’s value of teams and interactions over processes and tools, and the intent behind the concept is to empower teams to work together as they need to.

**How it looks in practice:**

* Self-organizing teams are autonomous groups within the organization who take control and responsibility over their respective projects and have ownership of those areas. Different organizations practice this principle differently. Spotify, for example uses “product squads” to practice this.

## Agile Principle 12

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

If you’re truly living by agile principles, there is no place for “we can’t change because we’ve always done it this way.” Just like we’re always learning new things about our customers and markets, we’re also learning from the processes we’re using to learn those things. Agile is not about following a strictly-defined process for every sprint and release, it’s about continuous improvement. And that continuous improvement must also extend to processes and teams.

**How it looks in practice:**

* Experimentation and testing is not limited to the product only. Agile teams are encouraged to experiment with their processes. You may think you’re already doing something well only to experiment with a revised version of the process and discover an even more effective method. Experimenting with your process and team is just as important as experimenting with the software you’re building.
* Regular [retrospectives](https://www.productplan.com/learn/product-retrospective/) are opportunities for the team to discuss what went well, what didn’t go so well, and where the process can be tweaked to improve things in the future. They’re an excellent place for product managers and product owners to learn if they are communicating effectively with developers and giving them the support they need before, during, and after sprints.
* Another consideration to make regarding this agile principle is that in order to practice it effectively you need to create a culture of trust and transparency that encourages openness and frequent sharing of feedback.