

## TABLE of CONTENTS

1. [Overview](#)
2. [Case study](#)
3. [Process](#)
  - i. [Ignition](#)
  - ii. [Inception](#)
  - iii. [Procurement](#)
  - iv. [Delivery](#)
  - v. [Landing](#)
4. [Primers](#)
  - i. **Agile**
  - ii. [Lean Startup](#)
  - iii. [Human-Centered Design](#)
  - iv. [Open Innovation](#)
  - v. [Modular contracting](#)
5. [About](#)
6. [Glossary](#)

# Agile

## Definition

When most people think of “agile” they tend to think of specific agile methodologies like Scrum or Extreme Programming. Agile is really a philosophical approach to software development centered around a few [core principles](#):

1. People over process
2. Working software (for example, delivered value) over documentation
3. Collaboration with users and customers
4. Responding to changes in requirements

Specific methodologies such as Scrum, eXtreme Programming, and in some cases Kanban, offer detailed implementations of the principles of agility. Agile development involves an iterative approach to software delivery, where software is built incrementally from the start of the project, rather than trying to deliver it all at once near the end.

## Key resources

- Learn more about Scrum at <https://www.scrumalliance.org>

## Key terms

- **Product Backlog**: a prioritized list of items for the development team to deliver. The most important items are shown at the top of the product backlog so the team knows what to deliver first. Items are often in the form of **user stories**.
- **Epic** : A large or high level user story that can be broken down into a number of smaller stories.
- **Product Owner**: The member of a scrum team who is responsible for what the team produces and the order in which it's produced. The Product Owner is charged with making sure that the team produces something that is of value to users and customers.
- **Retrospective**: At end of each sprint, the team holds a retrospective to reflect and adjust practices. Any team member can voice a problem or propose a solution.
- **Scrum Master**: The servant leader of the team who facilitates, removes impediments, and generally ensures that the team is working well without managing them directly.
- **Sprint**: A short period of time (usually two weeks) during which the team produces some items of customer value. Valuable feedback is sought from users and customers at the end of each sprint.
- **Sprint Planning**: The team's process of understanding and committing to a set of value to produce during the upcoming sprint.
- **Standups**: A short, daily meeting typically held standing up and face-to-face to encourage brief sessions. This is not a status meeting. It's a meeting for team self-organization around the work of the day. Team members plan for the most efficient and productive day for the team. Long answers and discussions should have follow-up in smaller groups after the standup meeting.
- **User Story**: A short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user

or customer of the system. They typically follow a simple template: As a , I want so that .

- **Velocity:** The sum of effort estimates associated with the user stories accepted by the Product Owner during a sprint.

## History

So what does skiing have to do with agile principles? Much more than you would think. The Agile Manifesto was created from February 11-13, 2001 at the Ski Bird Lodge in the Wasatch mountains of Utah (fun fact: other discussed choices for this meeting were Chicago, IL and Anguilla. Chicago was thought to be too cold without enough fun things to do and Anguilla too hard to get to). The Manifesto for Agile Software Development, more commonly referred to as the [Agile Manifesto](#), was born from this fateful meeting of the minds. The manifesto, in full, is as follows:

*We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*

*Individuals and interactions over processes and tools*

*Working software over comprehensive documentation*

*Customer collaboration over contract negotiation*

*Responding to change over following a plan*

*That is, while there is value in the items on the right, we value the items on the left more.*

This manifesto led to 12 principles that guide the agile development process:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

## Common questions about agile

### **Q: What's the difference between agile and waterfall?**

A: The so-called waterfall methodology is a plan-driven approach to software development that requires much analysis, planning, and risky project and product assumptions to be in place before doing actual development. It's inflexible, hostile to change, and has a long history of delivering poor project results. Agile is a new, empirically-based approach that emphasizes frequent interaction with users and an iterative approach to creating a customer solution. Agile embraces change during a project, lowers risk, and generally improves the return on investment of software development.

### **Q: What's the difference between agile and Lean?**

A: Lean manufacturing is the public version of the Toyota Production System, developed in the years following World War II in Japan with important contributions from American pioneer W. Edwards Deming. Lean is a method of organizational and process improvement aimed at removing waste from the process of building value for customers. Lean rests on the twin pillars of Respect for People and Continuous

Learning. Agile has its conceptual roots in Lean but focuses most specifically on software development and provides a framework for a different cultural approach.

**Q: What's the difference between Scrum or kanban or Extreme Programming?**

A: Scrum and Extreme Programming are two agile methodologies that provide very specific direction on how to build software in a way that is consistent with agile principles. Kanban is a method of process management and improvement that comes from Lean. Each of them provides a disciplined approach to the process of software development and all of them come from a different fundamental framework than waterfall.

## Deeper dive

- [Agile Scrum from Mountain Goat Software](#)
- [Scrum guides](#)
- [Scrum Training Series](#)