**Difference between Agile and DevOps**

The key difference between Agile versus DevOps is that Agile is a philosophy about how to develop and deliver software, while DevOps describes how to continuously deploy code through the use of modern tools and automated processes.

**Agile’s Manifesto**

Agile software developers behave in a manner that is consistent with the values and principles defined in the Agile manifesto.

Written and signed by software development leaders in 2001, the [Agile Manifesto](https://agilemanifesto.org/) defines the 12 principles and four fundamental values that Agile practitioners must embrace, including:

* Working software over comprehensive documentation
* Individuals and interactions over processes and tools
* Responding to change over following a plan
* Customer collaboration over contract negotiation

While it’s importance to the software development community can’t be understated, the Agile Manifesto is remarkably short in length. It’s less than 500 words in total.

The Agile Manifesto insists that the continuous delivery of software to the client is the highest priority. It also provides no concise guidance, nor recommends any specific processes to follow. It is purely a philosophical exercise.

**How do you define DevOps?**

In contrast to Agile, DevOps has no defining document. DevOps has no universally accepted definition.

It’s not even exactly clear when DevOps made its way into the public lexicon. Many point to a 2009 Velocity conference presentation by John Allspaw and Paul Hammond, entitled [10+ Deploys Per Day: Dev and Ops Cooperation at Flickr](https://www.youtube.com/watch?v=LdOe18KhtT4). Many also consider Gene Kim’s book [*The Phoenix Project*](https://www.techtarget.com/whatis/definition/The-Phoenix-Project) as a driving force behind DevOps adoption.

| **Agile versus DevOps comparison chart** | | |
| --- | --- | --- |
| Comparison | Agile | DevOps |
| Inception | 2001 | 2007 |
| Inventors | John Kern, Martin Fowler at al. | John Allspaw and Paul Hammond at Flickr, and *The Phoenix Project* by Gene Kim |
| Highest priority | Continuous delivery of software | Continuous deployment of software |
| Founding artifact | Agile Manifesto | 10+ Deploys Per Day conference session and The Phoenix Project |
| Implementation frameworks | Scrum, Kanban, ScrumBan, Lean, XP | CAMS, [CALMS](https://www.techtarget.com/searchitoperations/tip/Use-the-CALMS-framework-for-a-smooth-DevOps-transition), DORA |
| Alternatives | Waterfall | Silo based development and deployment |
| Team size | Used by small teams of 10 or less | Implemented as a company wide strategy |
| Scope | Focuses on the development of a single application | A corporate wide approach to software deployment |

In their 2009 Velocity presentation, Allspaw and Hammond describe how confidence in their software testing routines had created a great deal of trust between development and operations. As a result, the company implemented a process that would automatically deploy code into production over 10 times per day. To automate such a thing was considered earth-shattering at the time.

Many things have changed in the software development industry since 2009, but the cornerstones of DevOps continue to be:

* Trust and collaboration between Dev and Ops teams
* Heavy reliance on comprehensive software testing routines
* Integration of modern tools to simplify development and operations tasks
* Automation of deployment into production without any human intervention or checkpoints

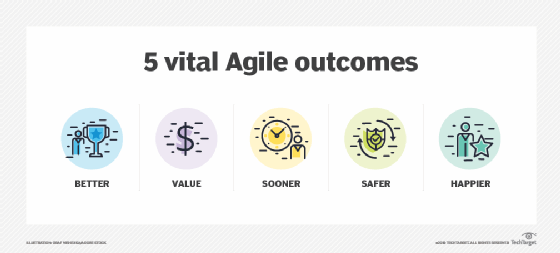
**Agile and DevOps culture**

While many argue over exactly what DevOps is, most definitions include the term ‘culture.’ Which begs the question: what is culture?

Generally speaking, culture is any group of like-minded people who use a common set of tools and follow a repeatable set of processes.

In essence, culture boils down to these three things:

* People:
* Processes: and
* the tools they use



Agile and DevOps goals and outcomes are closely aligned.

**DevOps processes and tools**

In the world of DevOps, practitioners stitch together many tasks and popular tools to accomplish them, including:

* code repositories and version control, such as Git and GitHub;
* containerization and orchestration, such as Docker and Kubernetes;
* automatic and continuous code development and integration, including [Jenkins and other tools](https://www.theserverside.com/tip/5-Jenkins-alternatives-for-Java-developers);
* Kubernetes for orchestration:
* [Static and dynamic code analysis](https://www.techtarget.com/searchsoftwarequality/tip/Static-and-dynamic-code-analysis-Complementary-techniques);
* Configuration management and infrastructure as code, such as Chef, Puppet and Terraform;
* Public cloud platforms and services, including from AWS, Microsoft Azure and Google Cloud

The overarching process that emerges from all of these steps is the continuous integration and continuous deployment of code (CI/CD).

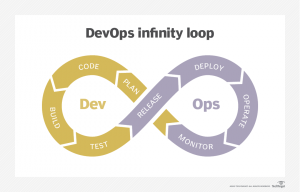
**How are Agile and DevOps similar?**

The path toward a successful digital transformation starts with embracing that Agile and DevOps can not only coexist, but they intersect.

All DevOps practitioners believe in:

* The automation of manual tasks and the importance of work not done;
* The importance of working with self-organized teams made up of motivated individuals; and
* The continuous delivery of software to the client as the highest priority.

Every single one of these points maps directly back to one of the 12 principles outlined in the Agile Manifesto.



The DevOps infinity loop shows the iterative dev process embraced by Agile teams.

**How to do Agile and DevOps together?**

Agile and DevOps practitioners share a common mindset. Moreover, their goals are aligned as well.

The ultimate goal of a DevOps transition is to fully automate the deployment of working code into production. This represents full DevOps enlightenment.

The Agile manifesto states in no uncertain terms that its highest priority is the continuous delivery of software to the client.

Both Agile and DevOps share the belief that the best way to build software, ensure transparency and promote sustainable development is to get working software into the hands the clients.

Agile and DevOps also intend for dev and ops teams to use modern tools and processes to get software into the hands of clients as soon as it’s humanly possible.

**Agile versus DevOps differences and similarities**

So to summarize, here are the most common differences and similarities between DevOps and Agile:

1. Agile is defined by the Agile Manifesto, while there is no universally accepted definition of DevOps.  
2. DevOps defines a work culture, while Agile is a software development philosophy.  
3. Agile’s highest priority is continuous delivery, while for DevOps it’s continuous deployment.  
4. DevOps insists on the automation of all manual tasks, while Agile values “the amount of work not done.”  
5. DevOps practitioners embrace an Agile mindset, while Agile demands participants be self-organized and motivated.

Together, people with Agile mindsets who embrace a DevOps culture all share a common goal: the continuous delivery and deployment of software to the client.

When all participants embrace an Agile mindset, DevOps-based digital transformations can be successful.