

# DEVOPS VALUES & PRINCIPLES



The IT industry is undergoing a major paradigm shift in the [software development process](#). Traditional software development methods have become obsolete thanks to:

- Increasing customer demands
- Rapidly evolving technologies
- Growing complexity
- Increasing security requirements

The shift towards the [software as a service \(SaaS\) model](#) and high dependency on [cloud-based technologies](#) has also contributed to the need for more agile software development processes. DevOps has become the solution to power fast-paced software development processes while providing the necessary flexibility.

*(This article is part of our [DevOps Guide](#). Use the right-hand menu to navigate.)*

## What is DevOps?

DevOps is the process of combining people, processes, and technologies to build higher quality software rapidly. The [DevOps model](#) combines the developers (Dev) with the operations (Ops) team rather than working as separate entities. This combination helps these teams to act as a single entity that manages the entire application lifecycle.

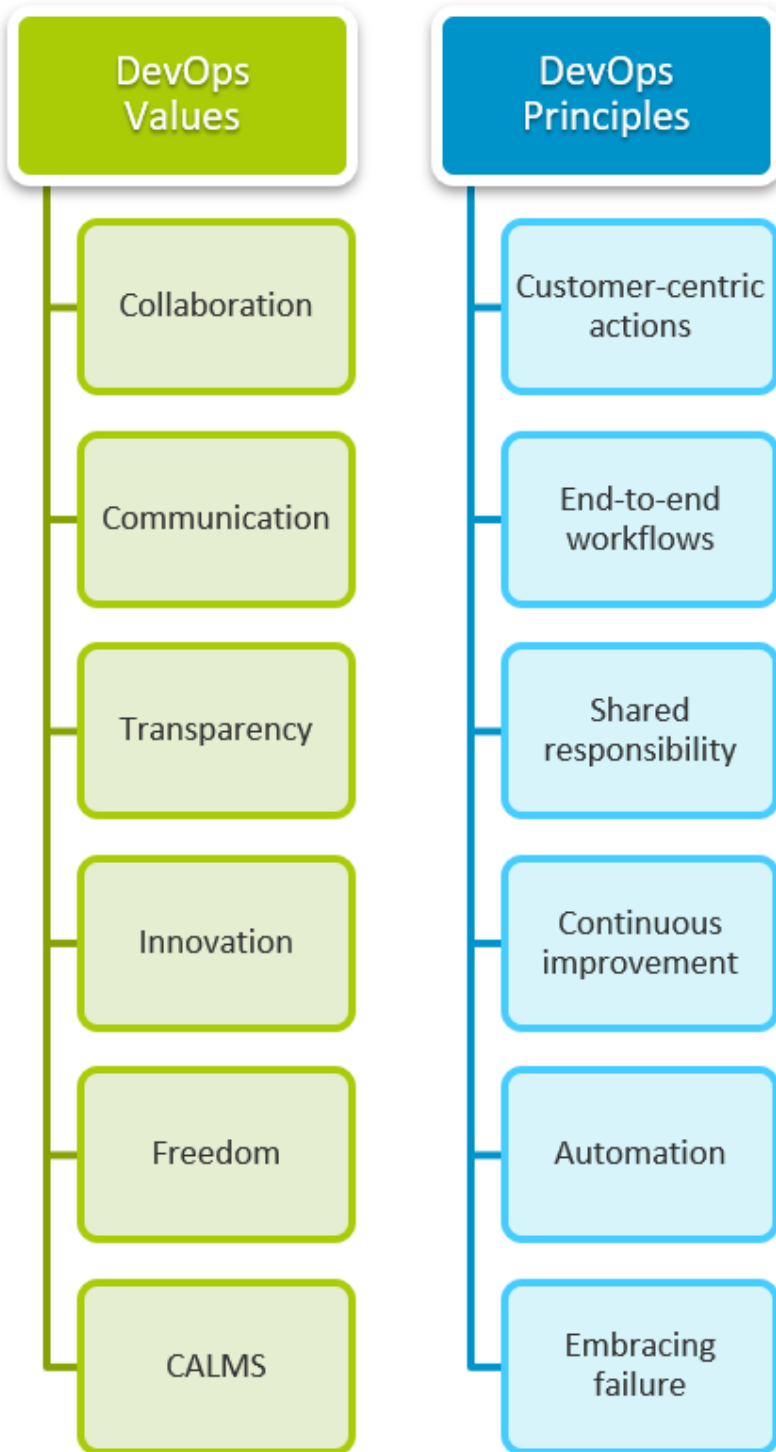


It has further evolved to include the security team, which has created [DevSecOps](#) that integrates security to all aspects of the software development lifecycle. All these things lead to a more collaborative culture across the organization.

Additionally, DevOps has paved the way for introducing automation to all aspects of the SDLC, from application builds to testing to deployment. With as a core tenant, DevOps aims to provide the necessary technology stack to create quality software efficiently while reducing errors.

Continuous Integration and Continuous Development tools are also a core part of the DevOps tool stack. The reason is that they act as the backbone of any DevOps process by providing a platform to integrate all the necessary tools and build automated processes.

Adopting DevOps helps organizations respond well to evolving market conditions and customer requirements while achieving their business goals.



## Core values of DevOps

The way DevOps is used can differ based on how it is implemented since there is no concrete value set that applies to each and every DevOps implementation. However, there are some core values that apply to DevOps regardless of the implementation.

### Collaboration & communication

When considering the universal core values of DevOps, the primary value is **collaboration**.

DevOps is focused on bridging the gap between different teams and creating a collaborative

environment where all teams are working together to benefit the product. This will lead to more **communication** between team members, allowing them to clearly understand each other's roles and responsibilities. It also helps them to understand how their work impacts other team members.

## Transparency, innovation, freedom

Open communications introduce more **transparency** within the organization. This includes stakeholder decisions, team activities, etc., which directly affect the end product. This transparency also eliminates ambiguity and allows organizations to move forward with clear goals in mind.

The freedom and flexibility offered by DevOps lets team members experiment freely, leading to more innovations. Encouraging **innovation** also plays a major role in the success of DevOps. With the **freedom** to innovate and experiment with team members, they can try out new technologies, methods, tools, etc.

These can then be implemented to improve the SDLC or the product itself.

## CALMS framework

All the values mentioned above should be applied as a fundamental cultural shift within the organization to get the most out of DevOps and implement proper DevOps practices. Using the CALMS framework is one of the ways to evaluate the progress an organization is making in adopting DevOps.

This framework analyzes the following aspects to evaluate the impact of DevOps in the business process.

- **Culture.** Embrace a culture of shared responsibility that includes all the stakeholders of the organization.
- **Automation.** Adapt automation in all the possible aspects of the SDLC
- **Lean.** Reduce [software development waste](#) with efficient resource usage and implementing best practices to create efficient processes.
- **Measurement.** Monitor every aspect of the SDLC, gather data, and evaluate to improve the DevOps process constantly.
- **Sharing.** Constant communication between teams to create a more collaborative environment.

By evaluating the above aspects, organizations can better understand their DevOps process and improve any deficiencies to align well with their business goals and core DevOps values.

## Principles of DevOps

Adapting DevOps can be a complex process, especially if you are moving from a more traditional software development model. Organizations can properly adapt or migrate to DevOps processes by focusing on the following DevOps principles.

## Customer-centric actions

Developments should target customer requirements with the primary goal of creating quality software faster. Customer feedback can be obtained easily with [shorter feedback loops](#) facilitated by the continuous incremental development process.

Then this feedback can be used to make the necessary improvements to the software or introduce new features to meet customer requirements. Developments can be planned to meet exact customer demands by following a customer-centric approach.

## End-to-end workflows

DevOps is a continuous process that encompasses all the aspects of the SDLC and does not stop after delivering the software. Software is continuously developed with regular releases.

This way, it is important to consider the overall development process when creating development workflows rather than considering an individual component. Workflows created considering all aspects of the SDLC create fewer chances for unexpected incidents that would affect the development process.

## Shared responsibility

In traditional software development, the responsibility of the software was divided [among different teams](#).

Developers were responsible for the development, while the operations team was responsible for deployments and maintenance. There were no shared responsibilities considering the product as a single entity shared by the organization.

In contrast, DevOps environments bring all these different teams together and share the responsibility of the software between all stakeholders. This allows [delivery teams](#) to work under a single directive developing and maintaining the software collaboratively.

## Continuous improvement

DevOps is constantly evolving. Therefore, organizations should continuously evaluate both the DevOps process and the software product even after creating a [solid DevOps pipeline](#). It helps to improve them to reduce waste, manage costs, optimize performance, etc.

This continuous improvement enables organizations to easily adapt to changing market conditions without having to undergo major cultural or technical shifts. When this is coupled with the freedom offered by DevOps to experiment and innovate, delivery teams can focus on improving products organically without making it another mundane task.

## Automate all possibilities

[Automation](#) is an integral part of any successful DevOps process. It can help save time and resources while organically increasing the efficiency of the SDLC. Automation should not be limited to repetitive manual tasks and should cover any task in the SDLC that can be automated from development to monitoring and support.

Even so, automation should not be done for the sake of automation if it does not offer any tangible benefit or may result in additional work that should be left out of the automation process.

(Compare [automation & orchestration](#).)

## Embrace failure

There will be issues along the way in any SDLC. However, failure should be considered a normal part of the development process to adapt to DevOps properly. This should come as an attitude change, with failures becoming learning opportunities to mitigate future issues and improve the product. One of the major barriers to innovation is that people are afraid that they will fail.

By removing such stigma, team members will be more willing to take risks and experiment which will ultimately lead to new innovations and improvements in the end product.

## What DevOps is—and isn't

DevOps is not simply adding CI/CD and automation to a software development lifecycle. It is a fundamental shift in how software is developed and managed as a whole.

Any organization can create a robust DevOps process to deliver software faster by adhering to the above-mentioned values and principles. It will also improve the overall software quality with fewer errors, making the software flexible enough to meet the ever-changing customer and market demands.

## Related reading

- [BMC DevOps Blog](#)
- [DevOps Culture & How To Successfully Achieve DevOps](#)
- [DevOps Job Titles, Roles, & Responsibilities](#)
- [Today's Top Trends in Software Development](#)
- [SRE vs DevOps: What's The Difference?](#)
- [The Cloud in 2022: Growth, Trends, Market Share & Outlook](#)