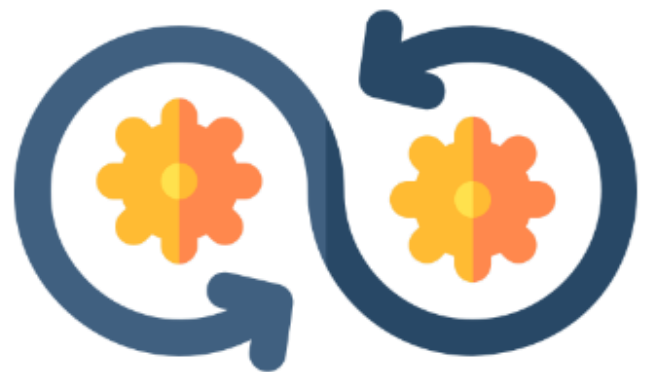
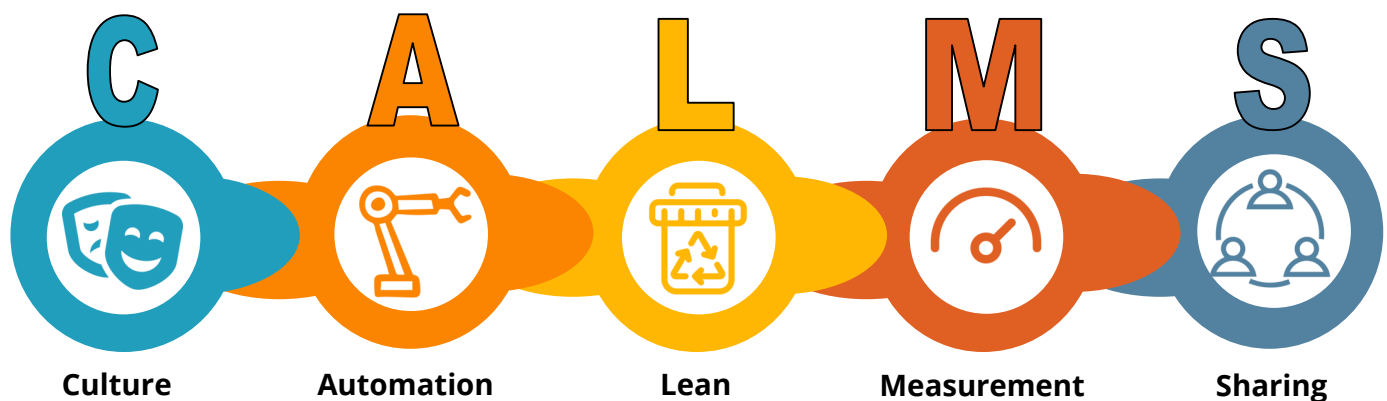


Best Practices to Implement DevOps



Introduction

Companies have been seeking new ways and methods in order to respond to the evolving customer needs. With that, DevOps emerged to sustain agility, speed, and innovation through redesigning the way employees and leaders work together. In fact, in the 2020 DevOps Trends survey, 99% of respondents said that DevOps had a positive impact on their organization. Furthermore, it has five pillars that influence how organizations are transforming themselves for the future, namely: (1) *culture*, (2) *automation*, (3) *lean*, (4) *measurement*, and (5) *sharing*. With the implementation of DevOps, there will be a shift in roles, change in desired skillsets, new incentives, and development of new policies.



A DevOps team has an engineering mindset focused on maximizing product delivery and customer value throughout a product's lifecycle. It consists of the following qualities:

- **agile:** focused on creating customer value through committed time and quality
- **self-organizing:** interdependent on other teams and has autonomy within established parameters
- **no sub-teams:** functions best as a single entity and does not split into smaller units
- **cross-functional:** continuous knowledge sharing and collaboration

8 Best Practices to Implement DevOps



Agile project management

Agile lets teams offer value to their clients more quickly and execute higher quality projects through an iterative approach to project management and software development. An agile methodology accepts that requirements and solutions may change as teams gather client input and gives the team the tools to respond to those changes.



Adopt Continuous Integration and Delivery (CI/CD)

CI allows integration of code changes more frequently. Error detection and quality issues are easier. On the other hand, CD gets the code from CI into production. Software deployments become low-risk events which can be performed anytime. The practice of continuous integration and continuous delivery (CI/CD), and deployment validate the ability to bring testing in the code deployment processes early.



Build with the right tools

Automation is a crucial part of a successful DevOps practice. With that, access and use of the right tools for each phase of the DevOps lifecycle is a requirement of the DevOps toolchain. These will improve software quality and speed of delivery.



Implement automation

Automating using CI/CD the building, testing, and deployment processes of software makes work easier for people who collaborate in a DevOps organization. End-to-end tests, unit tests, integration tests, and performance tests can all be made automated.

8 Best Practices to Implement DevOps (cont'd)



Monitor the DevOps pipeline and applications

One DevOps practice is continuous performance monitoring. It is crucial to monitor the DevOps pipeline so that any activity done will not cause unnecessary delays. Moreover, production applications must also be kept track of in order to identify failure or deficiencies in performances.



Observability

As monitoring is now considered to be more complex, an increasing interest in observability is perceived. Logs, traces, and metrics are the three pillars of information that must be used to make discoveries and predictions about the functioning of a complex system.



Gather continuous feedback

Customer satisfaction must always be the goal of an organization. With that, continuous feedback between teams is crucial in order for the members to have all the information needed to do work on a timely basis, and could make it possible to optimize for both speed and quality.



Change the culture

Establishment of collaboration, transparency, trust, and empathy in an organization culture are requirements of DevOps. These qualities allow teams to easily adopt DevOps practices. It is also important for different teams to have shared responsibilities and maximal cross-team communication and collaboration.