

A Quick Guide to  
**Transforming Software Delivery with DevOps**



**MENLO**  
TECHNOLOGIES

# Introduction

With the ever-increasing business consumption of IT services, IT organizations need a sustainable way to scale their service capacity. They need to manage the increasing demand while developing efficient, viable processes to safely scale.

But faster delivery of applications is challenging for both Infrastructure and Operations professionals as well as Application development and delivery teams.

Each are under pressure to increase speed without compromising quality. And with the advent of agile development and ever shorter release cycles and overlap between new development and maintenance releases, they must work together to find a common set of processes and tools that help them both.

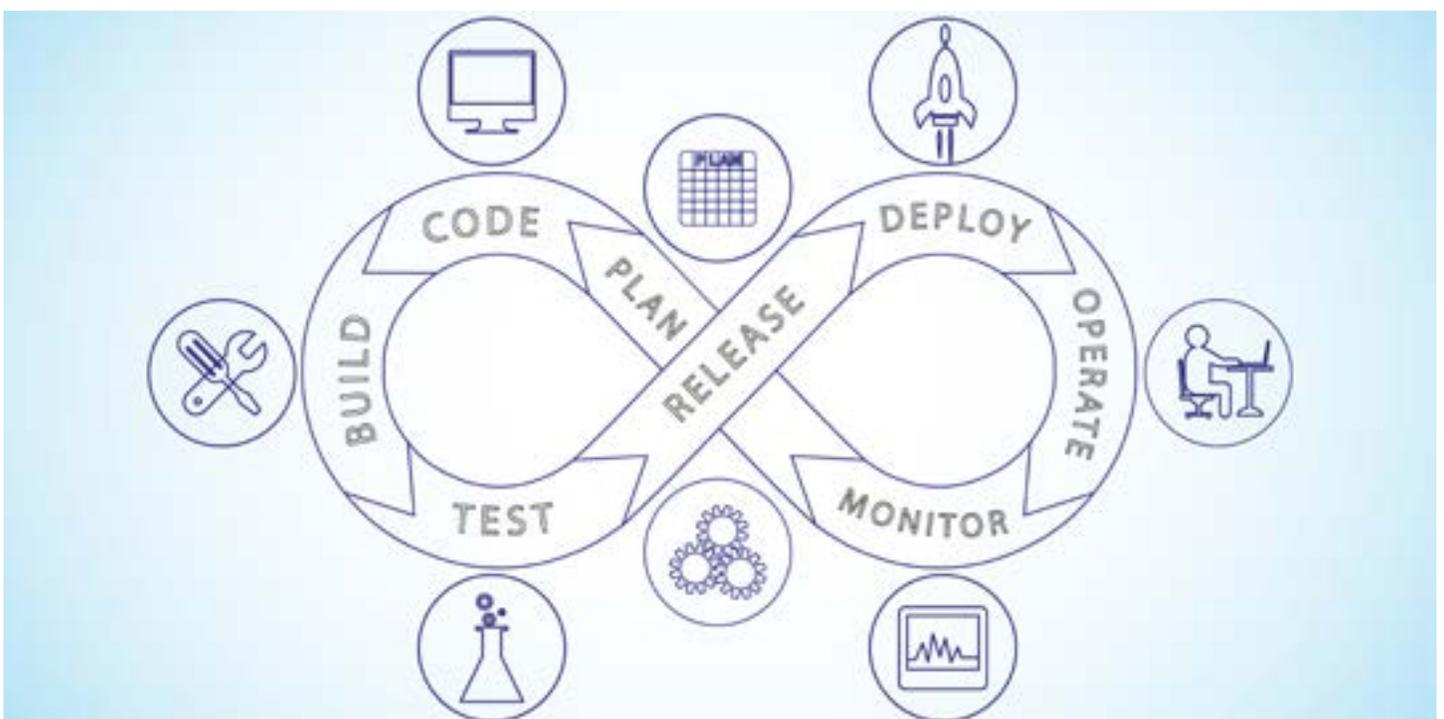
## **This is the core of the DevOps movement.**

Coined in 2009 by Patrick Debois, the term (development and operations) is an extension of agile development environments that aim to enhance the process of software delivery.

Debois observed that while developers want a fluid and changing system, operations want stability. DevOps is a response to the growing awareness that there is a disconnect between the development activity and the operations functions. Because development and operations want two different (and opposite) results, it creates what DevOps calls a “wall of confusion.”

DevOps reorganizes Software Development and Operations departments in order to remove the boundaries that currently generate delays in delivery of new features while improving software quality.

Menlo Technologies has prepared this quick guide describing how the DevOps methodology works, and how it has transformed software delivery. The guide answers the following questions we hear from our customers every day.



## What is DevOps ?

DevOps is an application development methodology that breaks down the barriers between Developer and Operations teams and leverages people, processes, and technology to stimulate collaboration and innovation across the entire software development and release process. DevOps practices include agile planning, continuous integration, continuous delivery, and monitoring of applications.

No matter what your organization does, IT is critical to your company's effectiveness. Whether they are client-facing or used internally, your applications are a critical element of business success. If your applications fail, so does your company.

## Why DevOps ?

Traditional IT development grew out of the waterfall method, which by its nature has a long development cycle. Developers were under considerable pressure to deliver substantial blocks of new features, their responsibility ending as soon as the software was handed to the operations teams to deploy. Operations teams were expected to keep the infrastructure as stable as possible, while deploying significant change in every infrequent release.

Since automated release tools were limited, operations were handled manually with a high incidence of mistakes due to human error. As a result, developers only released software a few times a year.

While a degree of specialization is essential in every company, it can create friction and an outdated structure.

Traditional development and operations teams tend to work in silos, limiting the amount of inter-team communication until software release times. This division seriously hampers speed, agility and effective coordination. Departments pull in different directions, communication is inefficient, and few people can see the big picture, which makes it difficult to resolve issues and ensure a holistic solution to challenges.



“DevOps is more like a philosophical movement, not yet a precise collection of practices, descriptive or prescriptive.”

[DevOps expert Gene Kim](#)

# What DevOps adds to the organization

Companies that incorporate DevOps practices get more done more effectively, plain and simple. With a single team composed of cross-functional members all working in collaboration, DevOps organizations can deliver with maximum speed, functionality, and innovation.

Technical benefits	Cultural Benefits	Business Benefits
<ul style="list-style-type: none"><li>- Continuous software delivery</li><li>- Less complexity to manage</li><li>- Faster resolution of problems</li></ul>	<ul style="list-style-type: none"><li>- Happier, more productive teams</li><li>- Higher employee engagement</li><li>- Greater professional development opportunities</li></ul>	<ul style="list-style-type: none"><li>- Faster delivery of features</li><li>- More stable operating environments</li><li>- Improved communication and collaboration</li><li>- More time to innovate (rather than fix/maintain)</li></ul>



DevOps team members build a culture of trust and cooperation which encourages them to improve the organizational products and services by continuously researching about creativity and innovation. These efforts allow the organization to better understand and address customer needs. The organization becomes more performance-based than power-based. This makes the workforce more creative and productive.

## DevOps Tools

To get the best out of DevOps implementation, you need to choose the right tools to enable automation across all aspects of Development, Deployment and Operations. According to a 2013 study by Puppet Labs, more than 80% of high-performing software-development organizations rely on automated tools for infrastructure management and deployment. You may already have many of the tools used in DevOps environments, but which tools are best suited for your organization's needs? Some well established, market leading DevOps tools follow.

“DevOps has many moving parts, and you need to have best practices and technology in place for each step.”

[David Linthicum, Cloud Computing Visionary](#)

## Jenkins

This tool automates the common developer tasks including compiling source code into binary code, creating executables, running tests, and creating documentation. Jenkins works with other tools such as Ant, Gradle and Maven to complete the job. Jenkins runs jobs on a scheduled or on-demand basis. With various plug-ins, you can also review code quality and build history.

## GIT

Git was created 10 years ago in response to the Linux community's need for Source Control Management (SCM) software that could support distributed systems. Git is probably the most popular source management tool available today. In addition to its great forking and pull request features, GitHub also has plug-ins that can connect with Jenkins to facilitate integration and deployment.

## Docker

Docker is the world's leading software container platform. Developers use Docker to eliminate "works on my machine" problems when collaborating on code with co-workers. Operators use Docker to run and manage apps side-by-side in isolated containers to get better compute density. Enterprises use Docker to build agile software delivery pipelines to ship new features faster, more securely and with confidence for both Linux and Windows Server apps. It eases configuration management, control issues, and scaling by allowing containers to be moved from one place to another. Docker automates the repetitive tasks of setting up and configuring development environments so that developers can focus on what matters: building great software.

## Ansible

Ansible is a configuration management tool that is similar to Puppet and Chef. Ansible provides configuration management, application deployment and task automation. It can also do IT orchestration, where you have to run tasks in sequence and create a chain of events which must happen on several different servers or devices. Ansible uses SSH, which is installed on all the systems you want to manage. It's written in Python which needs to be installed on the remote host.



# Understanding the relationship between DevOps, Agile and Lean

Are Lean, Agile, and DevOps distinct movements? Not really. Agile and DevOps share a common (Lean) goal, to improve the speed and quality of value delivery.

The difference between the three methodologies is in the range of focus:

Agile's focus is to optimize software development. Agile's small batch/iterative development improves work quality in development, while Lean helps that expand across the entire value stream.

DevOps recognizes that optimizing software development pushes the constraint downstream to Ops, so it works to break down walls between the two. DevOps uses Lean to improve collaboration, culture, and work quality.

Lean broadens the focus, focusing on optimizing the entire value stream. Lean concepts of continuous improvement, visual management, limiting Work in Process (WIP), and more to stay focused on delivering value to the customer.

Lean, Agile and DevOps IT principles are complimentary. DevOps stresses the importance of collaboration, communication, and convergence between development and operations. DevOps can use the benefits of Lean methodologies into operations by reducing the barriers to delivering more value to the customer (Lean) and aligning with the business. This results in a better running development team and smoother operations.



“A phased approach to continuous delivery is not only preferable, it’s infinitely more manageable.”

[Maurice Kherlakian](#)

# DevOps and Microservices

Microservice architecture is a way of designing software applications as a suite of independent deployable services that can scale and grow as needed. Microservices exist because of the need to deploy changes quickly, reliably, on an infrastructure with the lowest cost. It is an answer to the competitive demands for new features and better performance from users and businesses alike.

For those in the field of DevOps fields, microservices bring some significant benefits, including:

- **Deployability:** Microservices offer increased agility, which fuels the ability to roll out new versions of a service. That agility is due to shorter build, test, and deploy cycles. Microservices can also incorporate the flexibility needed to employ service-specific security, replication, persistence, and monitoring configurations.
- **Reliability:** A fault with a microservice only affects that microservice, and its consumers. When monolithic applications experience a fault, the entire monolith may fail.
- **Availability:** Releasing a new version of a particular microservice requires very little downtime, whereas rolling out a new version of a service in the monolithic application normally requires a full restart of the entire monolith.
- **Scalability:** Microservices can be scaled independently using pools, clusters, grids. That deployment characteristic makes microservices a great match for the elasticity of the cloud.
- **Adaptability:** Microservices offer the flexibility to consume new frameworks, libraries, data sources, and other resources. As loosely coupled, modular components, microservices prove to be easier to work with and support dynamic discovery and binding via a registry.
- **Management:** Microservices can leverage the agile methodology, where the application development effort is divided across teams that are smaller and work more independently.

Microservices bring additional productivity to DevOps by embracing a common toolset, which can be used for both development and operations. That common toolset establishes common terminology, as well as processes for requirements, dependencies, and problems, which in turn makes it easier for Devs and Ops to work with one another by allowing those entities to work jointly on a problem, and successfully fix a build configuration or build script. DevOps and microservices work better when applied together.

[Read more about Microservices Architecture.](#)



## DevOps and Azure

In a traditional organization, developers need to request each resource that they need from their IT Ops (Operations) team. Though DevOps adds velocity to the delivery process, the cloud environment enhances application development even more.

With a proper DevOps setup with Cloud based hosting, the DevOps team can spin the servers in the cloud and setup all of the necessary resources to set up the system up and make sure it is running efficiently. This Cloud environment is helpful to the new resources joining the team, and it's also beneficial to the organization in terms of cost cutting and maintaining infrastructure locally.

Cloud infrastructure also brings benefits to other teams. A QA team, for example, might have only a browser to test the application instead of setting up the whole system.



If you have a good DevOps approach, you can leverage the speed and agility of the Cloud without developers creating their own resources and tools that may compromise security or violate corporate regulatory and compliance obligations.

Azure streamlines the introduction of new computing and development resource. The cloud platform unlocks the productivity of your development teams, and simplifies the work of your Ops department. Instead of waiting weeks to get started on new development projects, your business can get to work in hours...literally.

The main obstacle while implementing the DevOps practice is, as we mentioned earlier, selecting the right tools. While there are many tools available in the DevOps ecosystem, picking the right ones for your organization can be challenging. You might overpay for the tools and miss your delivery goals or find that you are missing the most important tool that could really make difference in your delivery cycle.

The Azure stack provides a feasible solution for this problem. From the Microsoft Ecosystem for DevOps you can achieve all your DevOps needs. You can also plug in and use any third party tool as needed.

[Why Menlo Technologies recommends Microsoft Azure](#)

## Conclusion

According to the [2016 State of the DevOps Report by Puppet.com and DevOps Research and Assessment \(DORA\)](#), high-performing IT organizations employing DevOps practices are deploying 200 times more frequently than their low-performing counterparts. Correspondingly, these organizations are also able to enjoy 24X faster recovery from failure and minimize downtime.

Menlo Technologies DevOps Solutions will add value to your business by helping you ship code more quickly and reliably. Menlo's DevOps principles along with Azure bring agility to the business and improves the process as a whole.

## [Read Menlo's Guide to Microservices Architecture](#)

Menlo Technologies  
[www.menlo-technologies.com](http://www.menlo-technologies.com)

### CONTACT US

Menlo USA

4675 Stevens Creek Blvd • Suite 230 • Santa Clara, CA 95051 • +1 (408) 736 - 8100

Menlo India

4th Floor, A Wing, Pioneer Towers • Madhapur, Hyderabad 500081 • +91 (40) 6555 - 5762

