DevOps

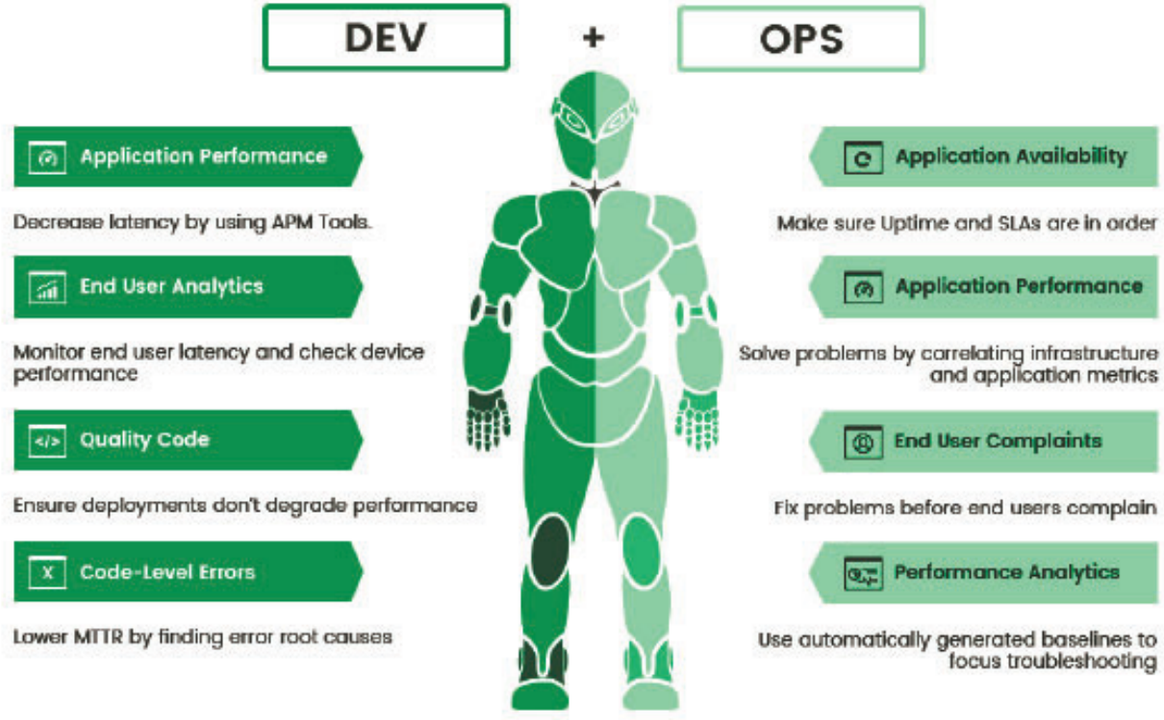
Faster deliver the value
of higher quality

serengeti
software tech

Introduction to DevOps

Traditionally, product development, testing, and QA were the responsibilities of three different organizational teams, grouped by specific expertise: developers, testing engineers, and the operations team. The responsibilities of each department have been precisely defined. Developers write code, testing engineers test the applications, and the operations team deals with the application in production. Although it can be perceived as a good thing, strict segregation of tasks can lead to weak collaboration and poor communication.

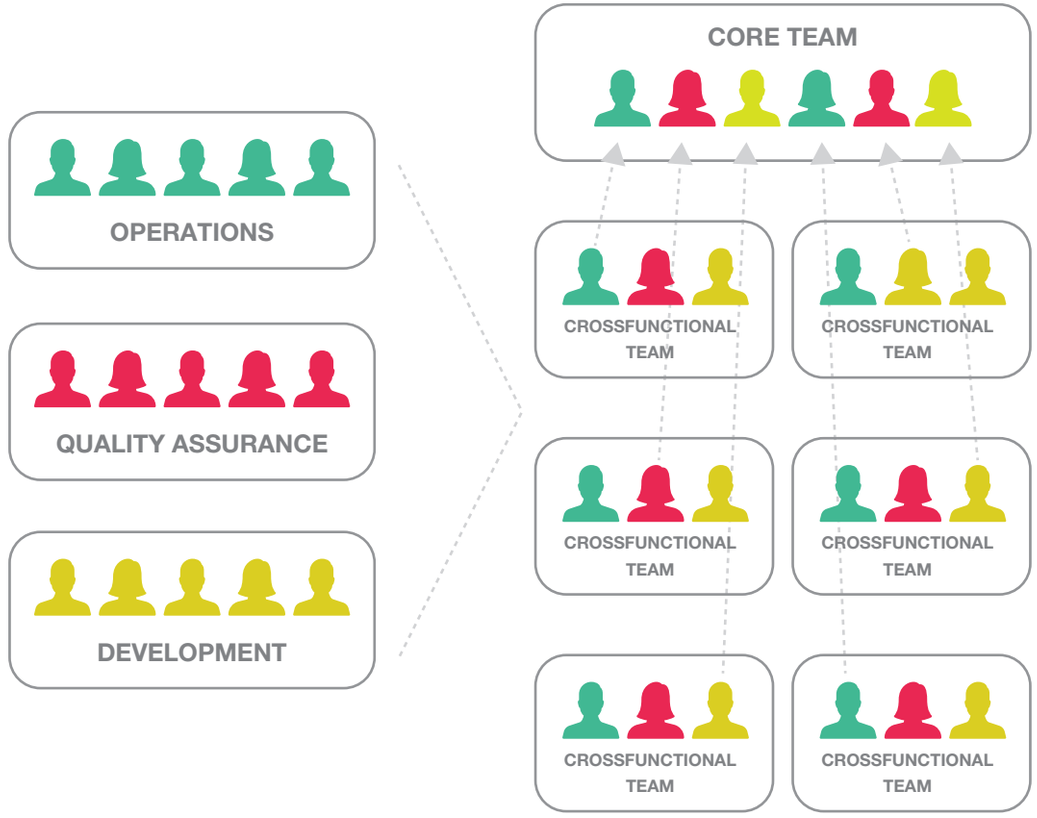
That's where DevOps comes in. As a set of principles that combines software development (Dev) and IT operations (Ops), DevOps focuses on improving communication, collaboration, and automation. DevOps can also be defined as a combination of flexible practices, tools, and mindset that increases the organization's ability to deliver quality products faster. Its popularity can be seen through today's buzzwords, such as microservices, cloud, and automation. When you talk about these things, you talk about DevOps.



Source: https://medium.com/clouddrove/how-to-become-a-devops-engineer-3705d1d0a9b9

Introduction to DevOps

The purpose of DevOps is to shorten the time between creating the change and the change being deployed to production. This is achieved through various tools, process automation, and eliminating complexity due to multiple teams by creating a single larger cross-functional DevOps team.

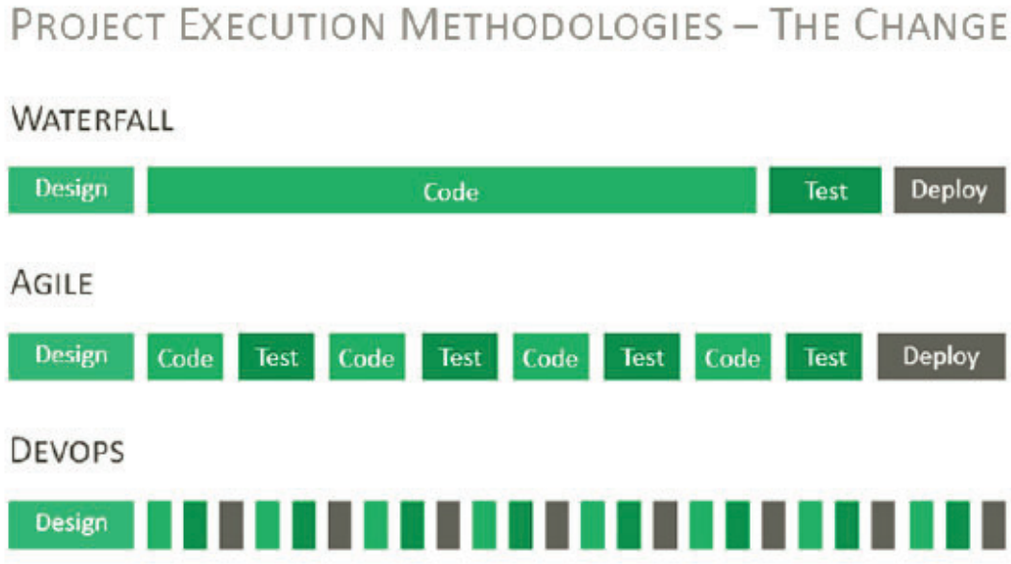


Waterfall, Agile and DevOps

The waterfall methodology defines the course of development which includes the development of specification, design, implementation, verification, and maintenance. A lack of flexibility to accommodate customer changes is often discussed as a disadvantage of this methodology.

To improve software development, companies started to adopt methodologies that focus more on customer collaboration and better communication between developers, customers, and users.

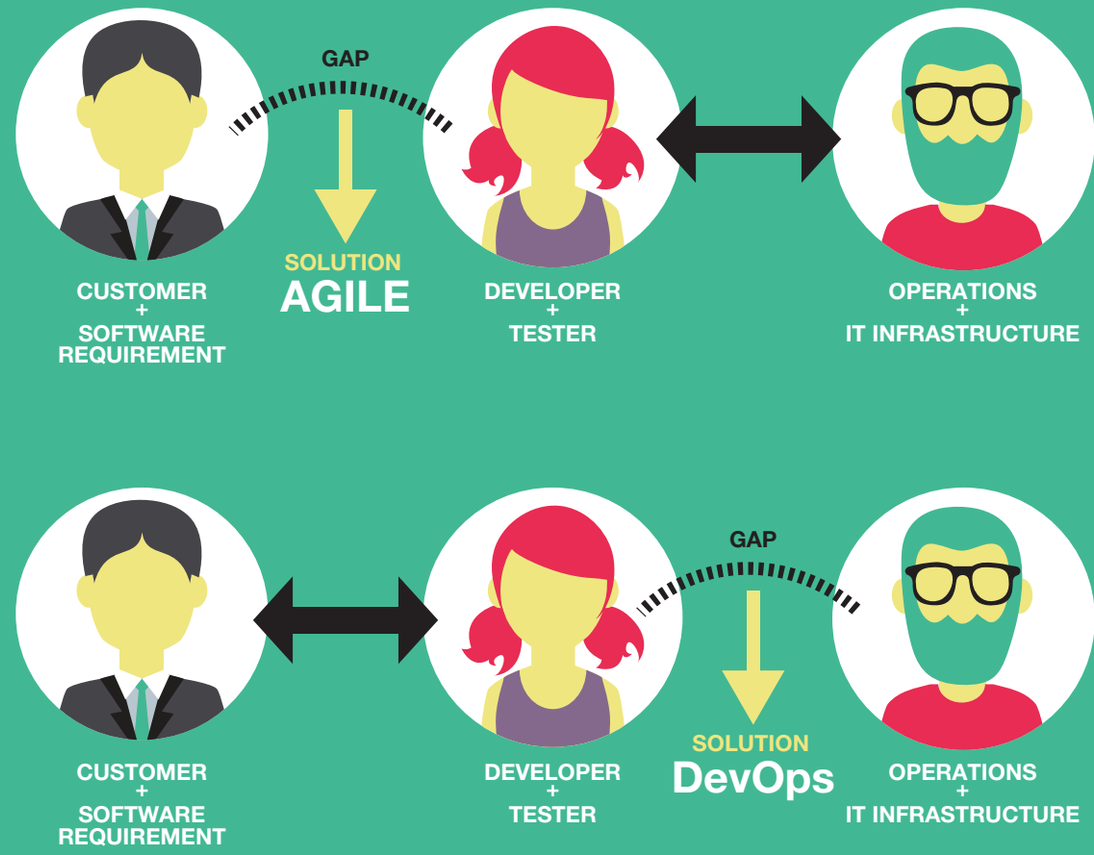
With agile methodology, flexibility in the software development process, collaboration and customer feedback have significantly improved. This methodology offered a shorter development life cycle and improved software quality because of the shorter iterations. The agile methodology addresses gaps between end-users and development. However, this methodology didn't solve the gap between development and IT operations.



Waterfall, Agile and DevOps

The gap between development and IT operations has been solved with DevOps. The application of the DevOps principles leads to one team (development and IT operations) working on the project, guided by one manager (DevOps leading role) as a bridge between development and IT operations. DevOps is not only compatible with agile; it can be defined as the automation of agile methodology. It applies agile principles beyond the development team.

DevOps focuses on constant testing and delivery. If you had to describe DevOps in one word, this word would be "continuous." With principles of continuous development, testing, integration, implementation, and continuous tracking during the lifecycle, the lines between development and operations are blurred and everyone is responsible for overall application performance.



Source: https://www.guru99.com/agile-vs-devops.html

DevOps - Faster deliver the value of higher quality

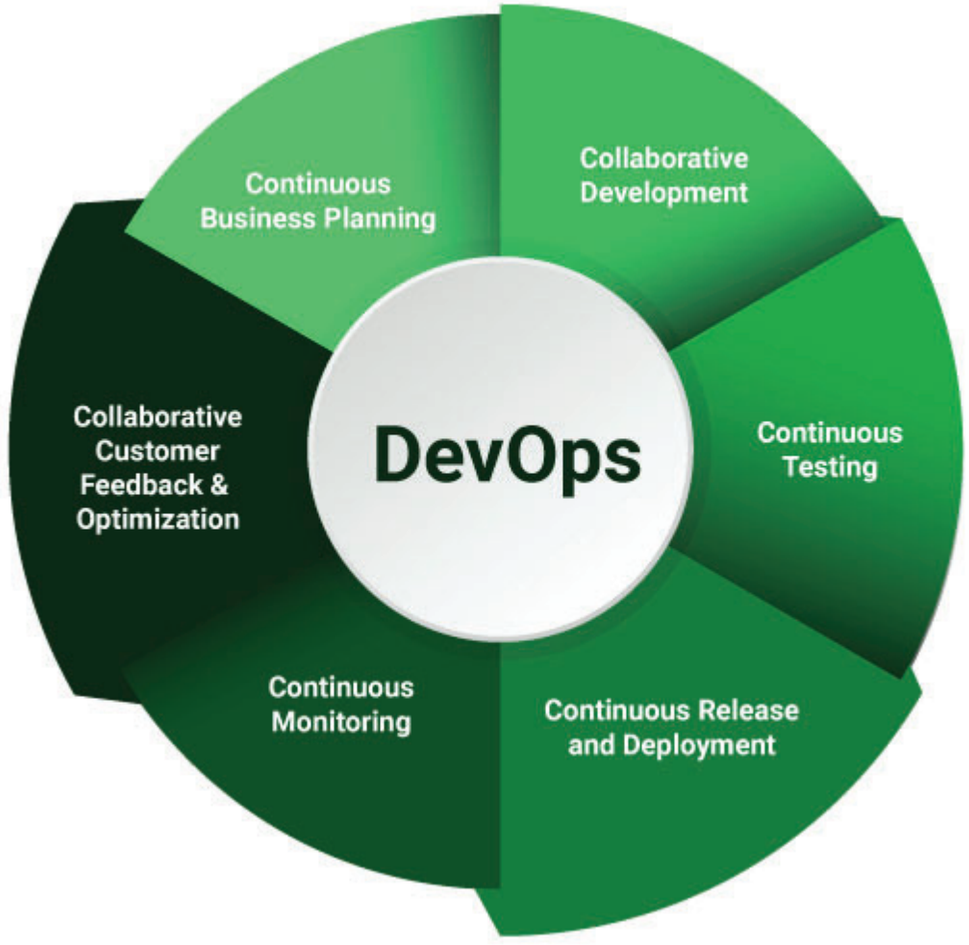
DevOps lifecycle

The DevOps lifecycle is a process, a journey of product development from start to finish.

It consists of:

1. **Continuous Development**
2. **Continuous Integration**
3. **Continuous Testing**
4. **Continuous Feedback**
5. **Continuous Monitoring**
6. **Continuous Deployment**
7. **Continuous Operations**

Every good project starts with a plan. Planning is a part of the first phase of the DevOps lifecycle, and it includes production and business metrics, release plans, security policy, etc. The second part of the development phase is developing the code, and it begins by choosing from different programming languages. This part includes designing the software, code quality, and performance, release candidate, etc.



DevOps lifecycle

The second phase is the core of DevOps principles. Modifications of the source code are frequent. They can happen on a weekly or even daily basis. With Continuous Integration (CI), code changes are built and tested automatically. Every time a team member commits changes to version control, it triggers an automated build system to grab, build, test, and validate the latest version of code.

Testing is also performed continuously. Continuous testing helps to find the risk earlier. It focuses on achieving better product quality.

When it comes to continuous delivery, feedback becomes an important part of release and deployment.

Continuous delivery allows faster and more frequent software building, testing, and releasing. Teams produce software in short cycles that enable continuous release without errors, thus delivering value to end-users.

In DevOps, monitoring is also performed continuously. It allows identifying the issues of specific releases. Unlike previous phases, IT operations are much more involved in continuous monitoring. Some activities related to this phase are end-user monitoring, different metrics, statistics, performance, etc. Monitoring increases situational awareness and enables performing corrective actions in time, often even before the customer notices a problem.

The next phase in the DevOps lifecycle is continuous deployment. During this phase, every change that passes the stages of production pipeline can be released, without any human intervention.
The last phase in the DevOps lifecycle is continuous operations, the purpose of which is to automate the release of the app and its updates.

Implementation

Align the organizational strategy and establish the vision and goals. In order to successfully build a DevOps culture, you'll need to have 3 components:

People – Collaboration is one of the key factors in DevOps. Team members must be on the same page, they must work together, communicate well, and focus on product quality and process consistency.

Process – To achieve an efficient and sustainable DevOps operational process, you need to assess the current DevOps maturity, define success factors, and establish a DevOps roadmap.

Tools – For every DevOps requirement there are numerous tools meant for facilitating the processes. Choosing the right tools can help you with improving the product quality, faster problem solving, improving productivity, continuous delivery, etc.

The first step in DevOps implementation is understanding what DevOps is. It's more than just tools and automation; it's a shift in organizational culture. A combination of practices and a mindset that increases the organization's ability to deliver quality products faster.

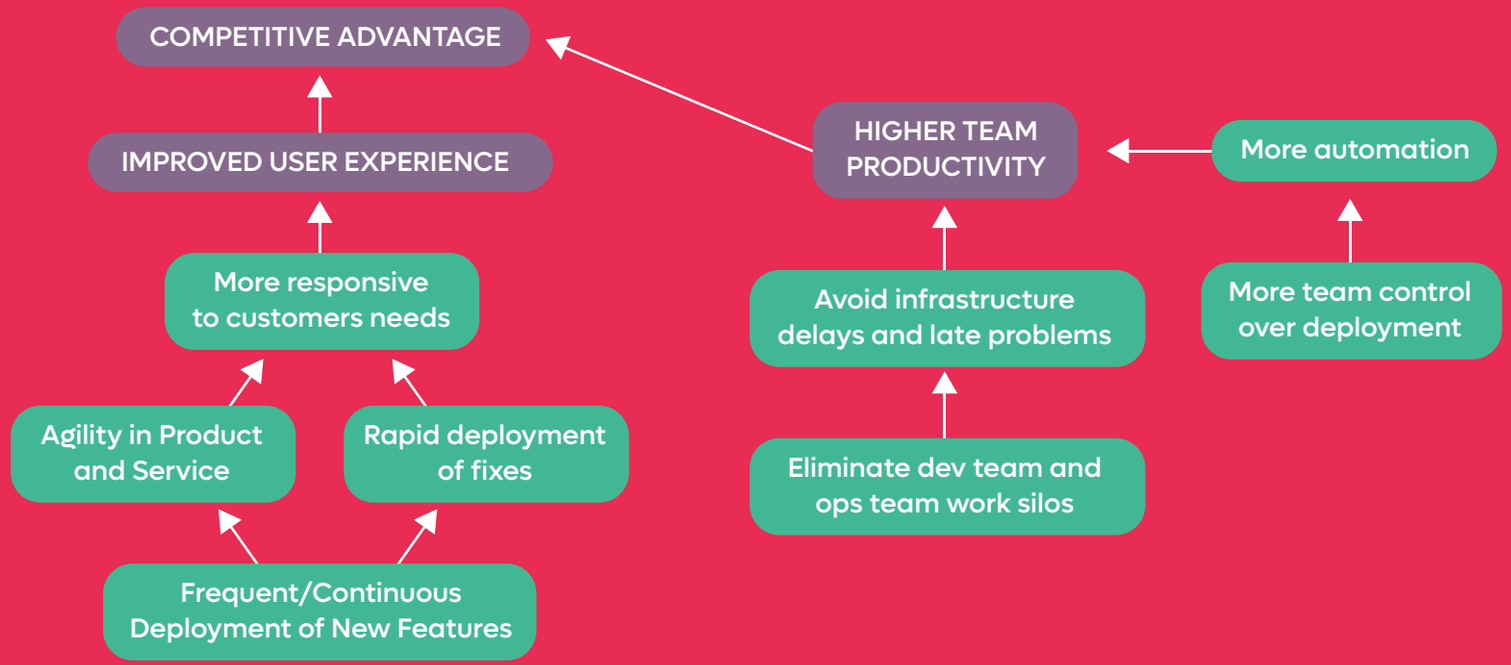
After aligning the organizational strategy, establishing a vision, and setting up goals, it is advised to assess the current DevOps maturity, to identify the main weak points. The next step is the prioritization of focused areas in alignment with previously set up goals.

The final step includes roadmap implementation and monitoring. Define what metrics are important and monitor them. For example, some metrics that can be tracked and monitored are deployment time, error rate, application performance, deployment frequency, etc.

Source: https://devops.com/implementing-devops-start-with-a-strategy/

Drivers for implementing DevOps

Transforming a traditional organization to adopt DevOps principles can be time-consuming and expensive. Nevertheless, a lot of organizations are implementing these principles, because the expected benefits from the outcome are greater than the cost.



DevOps - Faster deliver the value of higher quality

Drivers for implementing DevOps

More Time for Innovation

Implementing DevOps will significantly simplify certain processes and automate some of them. This leaves more time for brainstorming, creating new features, and getting new ideas.

Improved Team Collaboration and Communication

DevOps implementation leads to better mutual collaboration and improved communication. This especially goes for development, operations, and business teams. Previous strict boundaries between different departments are now blurred and team members share responsibility for product quality and success. Also, DevOps implementation virtually eliminates human error from the automated release- pipeline, which reduces stress and stabilizes the team.

Faster Deployment

Due to DevOps principles – continuous delivery, continuous integration, feedback cycle, the software delivery process is much faster, while product quality remains high.

Infrastructure as Code

DevOps is based on practices, technologies, and tools that make development, testing, and deploying faster. In the core of that practice is Infrastructure as Code (IaC).

IaC allows testing of applications at an early stage of development in a production-like environment. With IaC, Operations need to specify the stack once and deploy it later as many times as needed.

Benefits of DevOps

Faster Detection and Resolution of Issues

Constant monitoring, the collaborative approach, and continuous testing allow faster identification and resolution of issues.

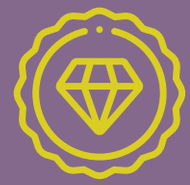
Improved Security and Stability

DevOps practices can reduce risks by eliminating environmental configuration differences. Additionally, because of automation, the risk of human errors is minimized. Continuous integration, test-driven principles, and automated deploys detect problems in the early product phase, so the issue can be resolved before it reaches operations.

Competitive Advantage with DevOps

Transforming a traditional organization to adopt DevOps principles can be time-consuming and expensive. Nevertheless, a lot of organizations are implementing these principles, because the expected benefits from the outcome are greater than the cost.

7 key benefits from DevOps



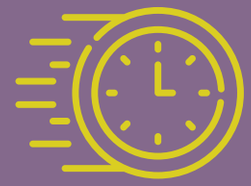
Significant improvement in the product quality

Improved preformance of the software

Minimum cost of production

Reilable infrastructure managed by coding

Improvement in productivity of the organization



Faster deployment

Faster problem-solving abilities

DevOps - Faster deliver the value of higher quality

serengeti
software tech

DevOps before & after

Before

- Delayed product deployment
- Miscommunication
- Separate workstream
- Integration in advanced phase
- Higher possibility of human errors

After

- Faster delivery due to continuous deployment
- Improved collaboration / teamwork
- Joint workstream
- Continuous integration
- Automation tools reduce chances for human errors

DevOps - Faster deliver the value of higher quality

Automation Tools

Using automation tools is one of the best ways to reduce human errors, improve quality, save time, and achieve better productivity. DevOps automation tools can be divided into two categories:

- **Tools by Functionality**
- **Tools by Requirement**

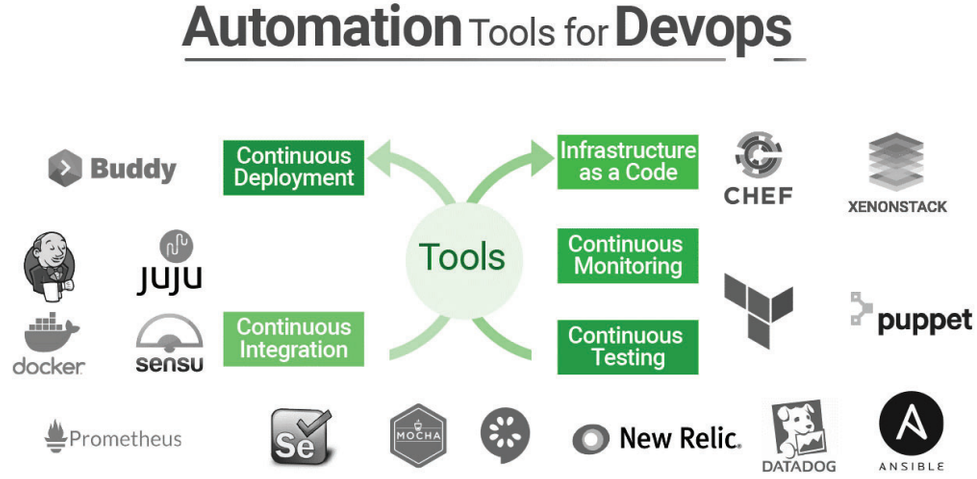
Docker is a DevOps technology suite. It allows DevOps teams to build, ship, and run distributed applications. This tool allows users to assemble apps from components and work collaboratively.

Puppet is a DevOps tool that allows managing entire infrastructure as code without expanding the size of the team.

Jenkins is a DevOps tool for monitoring execution of repeated tasks. It helps to integrate project changes more easily by quickly finding issues. It increases the scale of automation.

Terraform is a tool that maintains the state of infrastructure using 'state files.' Facilitates public and private cloud infrastructure provisioning.

CFEngine is a DevOps tool for IT automation. It is an ideal tool for configuration management. It helps teams to automate large-scale complex infrastructure.



Source: https://www.guru99.com/devops-tools.html

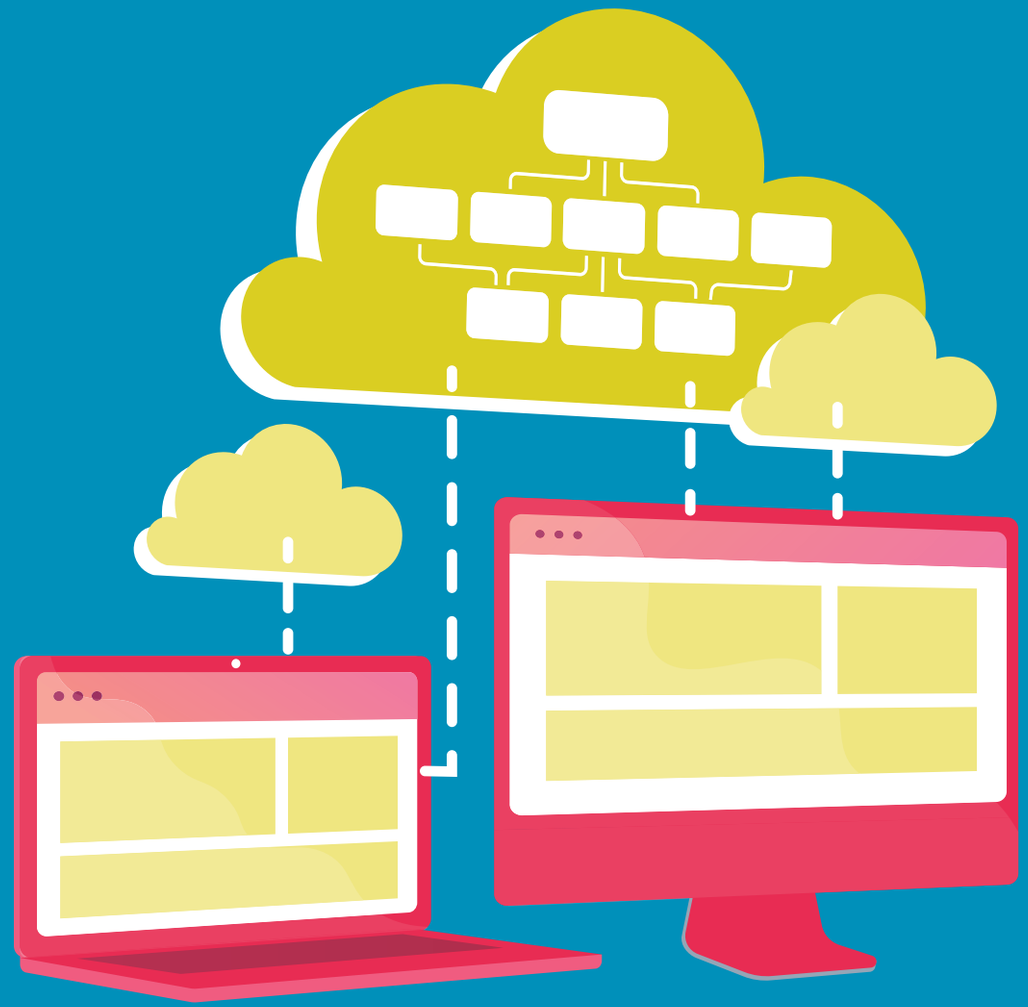
Cloud platforms for DevOps

DevOps relies on effective tools which automate manual tasks and help with complex environments. It's a cultural shift in how teams are structured and how they operate daily.

DevOps is not just a piece of software. But it is supported by technology that help the entire process running smoothly.

The first thing that comes to mind when talking about technology supporting DevOps is Cloud. Amongst other benefits, the Cloud allows easy building of test environments and enables automated testing in simulated environments. Some cloud platforms used for DevOps are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud.

These platforms provide a set of services designed to enable rapid and reliable building and delivering products using DevOps practices.



What about other industries?

Usually, when we hear DevOps, we connect it to the tech industry. However, DevOps practices can be applied to different, non-tech industries. Here are some of them.

Insurance
Nationwide started using a DevOps approach to application development and delivery. They've noticed a 70% reduction in downtime and improved software quality by 50%.

Source: https://www.ibm.com/downloads/cas/E3JAG4GX

Banking
Standard Bank switched to a DevOps practice because they were facing a growing demand for digital banking solutions. Now, Standard Bank has over two-hundred teams using a DevOps toolchain and have seen a 90% reduction in turn-around time.

Source: https://www.infoq.com/articles/state-devops-banking/

Why DevOps?

Implementation of DevOps brings numerous advantages, has a positive effect on team members, and speeds up product delivery while maintaining its high quality. Al so, it can boost your market competitiveness.

Over time, DevOps has proven to be more than a passing trend, as evidenced by the allocation of numerous resources by large companies such as Microsoft, Amazon, and Google to build tools and cloud platforms that support DevOps practices. Organizations want to take advantage of fast application creation and delivery, better performance, and a stable operating environment. They want to create innovations and be more competitive in the market.

Although it takes some effort to implement DevOps, the expected benefits from the outcome are greater than the cost.

Who are we?

An international nearshoring and offshoring software development and consulting company. With our unique Team Extension Model, we help companies with each stage of the development lifecycle, not only with development but also with consulting services. Technical consulting in different domains like Architecture Design, Automated Testing, DevOps and SCRUM was recog-nized in 300+ projects across different industries such as finance, logistics, hospitality, industrial manufacturing, healthcare, energy, and retail.

Serengeti d.o.o.
Ulica Pile I. br. 1
HR – 10000 Zagreb, Croatia
sales@serengetitech.com
serengetitech.com

serengeti
software tech