DevOps Assessment

Free Checklist to analyze the state of DevOps in your company

XALT

Preface / Introduction

The DevOps Methodology and Culture has taken hold in IT, Operations and even on departments that necessarily don't connect directly with software development. For a couple of years now, a growing number of organisations across the globe are adopting DevOps to deliver applications, services and other digital products at high speed.

By implementing DevOps in their organization they enabled themselves to ship digital products faster, evolve and update existing products more efficiently and frequently than other, traditional software development and infrastructure management environments.

After realizing, that the benefits of DevOps have not only tremendous impact on business success, but also brings teams of different departments together.

Implementing DevOps has been a game-changer for many organizations all over the world.

The questions though remains, how do I start implementing DevOps and if I have already started, how do I identify and map the areas where I need improvement?

Having a clear understanding of the maturity of my IT Infrastructure, Software development and culture will help you advance efforts greatly.

This assessment will help you to analyze how far your organisation has come in implementing DevOps practices. You can use this checklist as a self-assessment of your status quo and how you can further advance the level of maturity for each section.

DevOps Advancement Analysis

On the following pages, you will find sections of common characteristics, capabilities and general factors that can be found in organizations that have implemented DevOps.
For each section and statement you can score yourself between 0 (Not implemented) to 5 (Implemented throughout the organization). To get useable results we encourage you to score each statement truthfully.

Scores:
0 = We do not do this / Not implemented
1 = We rarely do this / Somewhat Implemented
3 = We do this to a moderate amount / Implemented this partially
5 = This is done/implemented throughout the organization

Agility and Velocity

To stay ahead of the competition and complete projects faster, speed and velocity to drive projects further are essential in today's world. Think about your development processes, do they speed up or slow down new releases?

1.	We ship a new iteration/version regularly within 1 - 2 weeks.	
2.	Small code changes are deployed quickly and securely without severe interactions with stakeholders.	
3.	We use agile processes and methods and have moved away from waterfall methods in software development.	
4.	We use retrospectives regularly to discuss what's working well, what isn't working at all and what we can improve. Top improvements are implemented within a short period.	
5.	We take action to ensure that the team doesn't run into bottlenecks.	
6.	We work on high-priority tasks in the backlog / Prioritize tasks.	
7.	Projects are moved forward even if the security scans haven't been completed yet.	

Collaboration and Culture

Analyze how effortless it is for teams of different departments (e.g. development, operation, security, and product management) to work and collaborate together across tools and methods used by each department.

1.	Knowledge and information are shared across teams within the organization.	
2.	We have implemented a centralized hub to store and retrieve information (e.g. Confluence, Sharepoint, Notion). This enables our teams to collaborate with people outside the team.	
3.	Team members across departments can meet face to face.	
4.	Communication, collaboration and transparency are easy to achieve across the entire project team. Silos have been broken down.	
5.	Collaboration between team members mainly focuses on retrospectives, troubleshooting and improvements. (Other topics are typically cleared out beforehand.)	
6.	Our team consists of a cross-functional delivery team of development, testing and operations.	
7.	We implemented a culture of experimentation and innovation.	
8.	We enable our team to work autonomously and independently with a high trust culture enabled.	
9.	The DevOps culture has been implemented across the organization.	

Automation

Automations are an essential part of your DevOps efforts. They speed up recurring tasks and free up important resources to be used on high priority tasks. Ask yourself, how automated is your DevOps pipeline? Are tests automated? Are security scans performed automatically? Is your infrastructure called by code?

1.	Security scans are automatically performed on every code change.	
2.	Scan results automatically issue a notification to the person in charge and create a work ticket in your project backlog.	
3.	Security scans rarely need manual interaction.	
4.	Is the provisioning, configuration and management of infrastructure (e.g. networks, storage, etc.) automated (for example by using Infrastructure as Code)?	
5.	We have automated configuration and management for the following environments:	
	Virtual Machines	
	Networks	
	Operating Systems	
	Applications	
6.	Provisioning, configuration and management of environments are automated.	
7.	**Automated** - unit tests (testing individual modules), - integration tests (testing the interaction of modules with each other), - system tests (confirming overall system functionality meets requirements), - performance tests, and - quality of code tests **have been implemented.**	
8.	We enable our team to work autonomously and independently with a high trust culture enabled.	
9.	The DevOps culture has been implemented across the organization.	

Architecture and Design

Research from the DevOps Research and Assessment (DORA) team shows that architecture is an important predictor for achieving continuous delivery. The right form of architecture is one of a set of capabilities that drive higher software delivery and organizational performance.

1.	The application is built from several stateless components with scaling and resilience provided at the application layer (full MicroServices).	
2.	There are fully separate development, test and QA environments	
3.	Our architecture enables the team to deploy services independently and every functionality can be updated individually and applied to the application.	
4.	All application logs are written to a central log repository automatically with the configuration included in the environment design to allow portability between environments.	
5.	It is possible to roll back the application when a critical error appears to the latest stable system. (e.g. by using the Blue-Green Deployment Method)	

Process efficiency & DevOps Practices

Think about the processes in your teams. Did everything run smoothly when your team just released new software or code? Was the project delayed given the used processes and practices?

1.	DevOps practices such as continuous testing, development, integration, delivery or deployment haven't been implemented	
2.	A defined code review and approval process exist in my organization.	
3.	An automated build of the software is triggered into a production-like environment, automated tests are then triggered and software is available to be automatically deployed into production	
4.	We implemented notifications to communicate build status and failures to the team.	
5.	We have implemented a standardized process that tests at least 90% of all code before reaching the security team.	
6.	We have implemented standardized test frameworks that are used by the teams.	

Security Culture

What is your organization‘s approach when a security breach occurs? Have all teams and stakeholders access to security guidelines and policies? Have all team members received a thorough security education? Do you empower developers with the necessary tools to create and deliver secure code?

1.	There is an awareness of security among employees (especially those outside the security team).	
2.	Incorporating security practices, like testing and code review, into daily work is empowering for employees.	
3.	It is the employee's responsibility to assess and maintain the security of their work environment.	
4.	A company's security policies are communicated clearly, are enforced by default, and are communicated on a regular basis.	
5.	Experts in cybersecurity evaluate and set security standards and automate their use whenever feasible.	
6.	Compliance is monitored and exceptions are investigated regularly.	

Calculate your results

Add up your scores from each section of the DevOps Assessment.

Total Score	Level of DevOps maturity
0 to 120	Beginner
121 to 180	Intermediate
181 to 225	Advanced

Next steps: Analyze the results and build an action plan to improve your DevOps advancements

DevOps is an ongoing process and something that can't be implemented just within a few months. It also offers the possibility to be improved over time and time again. For your team and organization to advance to the next level, analyze each section thoroughly where you scored low, and create a list of measures that help you and your team to advance further and improve your scores.

An overview of each category to achieve DevOps excellence

Each section provides a challenge for itself. But understanding the results of each category is essential to help your organization and teams reach your desired goals in terms of implementing DevOps. Each explanation for each category will feature beginner, intermediate and advanced scenarios, needed capabilities and a list of actions to reach a new level in DevOps.

Automation

To shorten the lead time for software delivery, automation is essential. If quality and speed are going to improve, automation is necessary. Automation allows environments to be provisioned and configured in the same way each and every time. With automation, you can build, test, deploy, provision, and configure application code easily. Automation enables you to monitor environments and respond to incidents (including security threats) based on a set of rules. With automation, consistency is ensured, quality is enforced, waste is removed, and speed is delivered; it is the technical heart of any DevOps implementation.

Levels of Maturity

Beginner

Teams have started to implement first automation for tests, provisioning and release management. Scans may be kicked off manually or via a custom script. But test results do not automatically trigger any action or remediation that helps teams to simplify recurring tasks. Time-saving in this stage is minimal and automatic security scans are still being conducted manually.

Intermediate

Automations have been implemented on a larger scale and are kicked off manually or by using a custom script. Automation of the provisioning, configuration and management of environments is done by tools. Therefore, a fixed set of tools is used by all stakeholders to automate various parts of development, operations or security.

Advanced

Automations have been implemented throughout the organization and free up vital resources to be used for more important tasks. Automated testing enables the teams to run tests quickly and frequently. Time savings through automation has been fully achieved.

Actions to take

1. Start by gradually automating the most important and time consuming, yet recurring tasks. These can be, configuration and management of environments or infrastructure.
2. Automate tests for every aspect of development, infrastructure and security aspects.
3. Determine which security policies can be best automated and build them.
4. Generate automatic notifications and tasks when tests, builds fail.
5. Use “as code” for everything. E.g. Infrastructure as code.

Architecture and Design

Development and architecture assess how well your designed workflow and development tools adhere to known good practices. The agility and functionality of Public Cloud environments (e.g. AWS, Azure) make them ideal for building DevOps-focused architectures.

Services such as the public cloud, help to simplify provisioning and managing infrastructure, deploying application code, automating software release processes, and monitoring the performance of applications and infrastructure.

The goal of DevOps is to increase an organization‘s efficiency and effectiveness by combining cultural philosophies, practices, and tools that enable the delivery of applications and services at high velocity: It enables companies to better serve their customers and compete more effectively in the market by evolving and improving their products more quickly than organizations that use traditional software development and infrastructure management processes.

The action here is simple. Start moving from on-premise to the cloud. Scaling, controlling costs and managing your infrastructure, architecture and overall design will never be easier.

Process efficiency & DevOps Practices

The concept of DevOps is an approach to application development and software release that enables rapid development and faster delivery of new or revised features.

By incorporating DevOps principles, application development teams and IT operations teams can have more seamless communication, collaboration, integration, visibility, and transparency.

In DevOps there is an ever-increasing relationship between developers and operations, from the initial software planning phase through to the development, build, test, and release phases, and on to deployment, operations, and ongoing monitoring.

Levels of Maturity

Beginner

Every project is run or secured differently. Although there are some standard tests, not all developers/security team members need to run them for every piece of code. A learning platform is in the works, but no employees are currently able to use it.
Further, standard DevOps practices like continuous testing, development, integration, delivery or deployment haven‘t been implemented yet.

Intermediate

Standard approaches of running projects and securing projects are in place and tests have been implemented that are an important part of the workflow. To some extent, DevOps practices have already been implemented. A learning platform has been developed and started to be used more over time.

Advanced

All projects follow a set of standardized procedures to ensure projects are run following the same approach, security and compliance are a vital part of releasing new code.
DevOps practices have been implemented on a wide scale in the development process.

Actions to take

1. Establish a learning platform for all teams to use
2. Implement standard DevOps practices (CI / CD etc.)
3. Standardize processes to simplify and streamline the workflow of all teams.

Security Culture

Traditionally, security has been viewed as a bottleneck rather than an enabler. The security team will be responsible for early maturity practices, however advanced maturity practices make security an organization-wide responsibility.

To encourage employee buy-in, the culture shift must come from the top of the organization. It is vital to let all team members know the urgency of strong security practices.

Generally, a highly advanced organization will have clear and specific policies, and its employees will be empowered to integrate security into their own work. Developers should receive scan results automatically in DevOps to enable them to identify and fix vulnerabilities without needing to seek additional reviews from security teams.

Levels of Maturity

Beginner

Typically, security is dealt with by the security team. Others feel that it is not their responsibility to maintain the security of what they work on. For them, security is a hindrance. Although policies or guidelines may exist, they may not be strictly enforced.

Intermediate

There is an understanding of the importance of security in the day-to-day work of employees, but they lack the necessary tools to improve the security of what is produced. Security messaging is reinforced internally, and guidelines, policies, and/or requirements for security procedures are typically followed and enforced. Managing security and improving it are intertwined in an uneasy tension.

Advanced

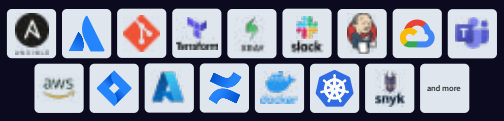
It is almost universally recognized that security is critical, and most believe they have the authority to incorporate security practices into their daily work. For all employees, security is a top concern. We communicate and enforce company policies clearly and regularly.

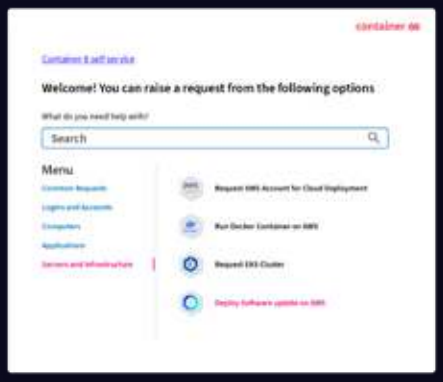
Actions to take

1. Establish a learning platform for all teams to use
2. Implement standard DevOps practices (CI / CD etc.)
3. Standardize processes to simplify and streamline the workflow of all teams.

Container8

The all-in-one DevOps as a Service Platform





Imagine an automated DevOps platform that has a low dependency on your infrastructure, is extremely unblocking, and provides the full industry-standard toolset to make DevOps easy.

A platform that follows the best practices to allow you to release faster and more often while saving time and money on tools, offers a simple onboarding experience that allows teams to run their own automation pipelines and adapt the platform based on their unique needs, is secure by design and transparent by nature.

Learn more about Container8 and how it unlocks the potential of DevOps.

Learn more

Contact Information

xalt.de info@xalt.de +4989416124240

XALT Business Consulting GmbH, Tomannweg 3, 81673 München, Germany

XALT