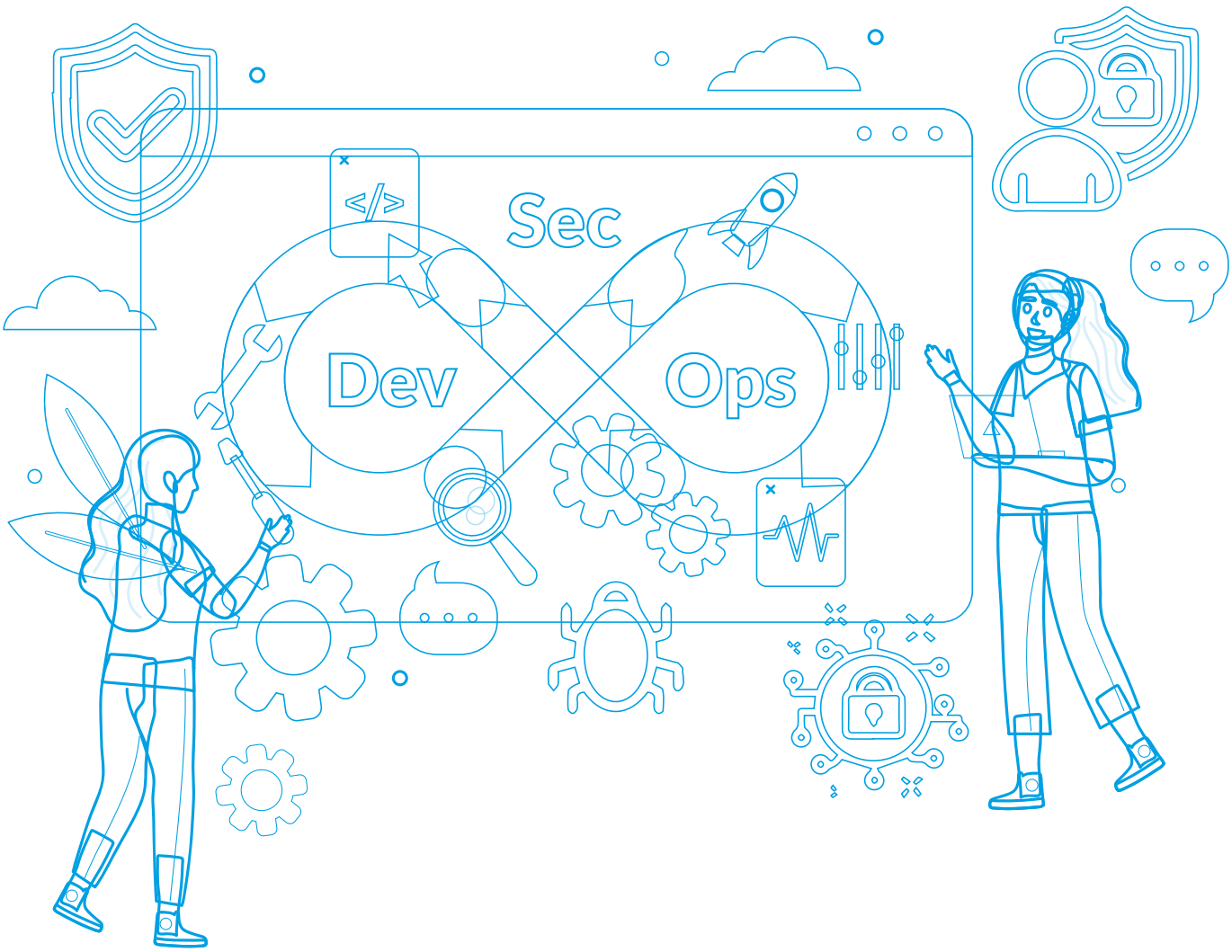
CODIFICATION

Complete Guide to DevSecOps



September 2022

Field Guide

Table of Contents

Introduction

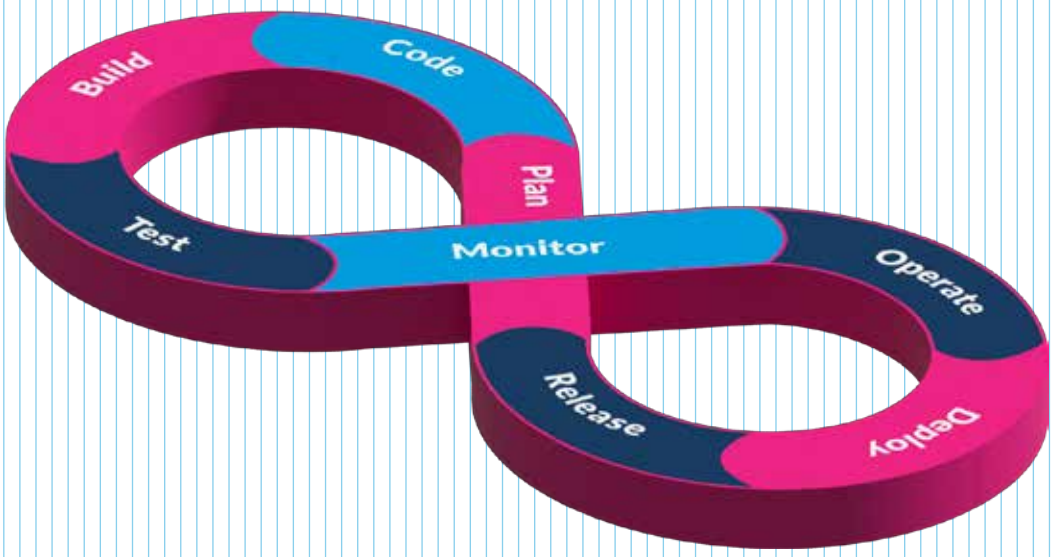
As part of the DevSecOps method, security testing and protection are built into every step of creating and deploying software. DevSecOps is a similar concept to DevOps in that it is as much about shared responsibility and culture as any particular technology or method. DevSecOps aims to provide better software more quickly while also improving the speed and efficiency with which production-level software problems are found and addressed.

The subject of DevSecOps can be a handful to digest. In this guide, we will break down each of those ideas in the parts that follow, so you can see how your company can embrace DevSecOps more comprehensively.

What is DevSecOps?

Development, security, and operations are three distinct disciplines that are connected tactically by DevSecOps.

The objective is to easily include security into your continuous integration and delivery (CI/CD) workflow in both development and operational settings (pre-production). Let's examine each discipline and how it contributes to producing stronger, more secure software more quickly.



Development

New software applications are created and improved by development teams. This comprises:

- Custom, internal applications created with a particular, focused goal in mind.
- Links are powered by APIs that connect modern services to ancient systems.
- Applications that use open-source code to hasten their development.

Modern development processes use Agile models, which emphasise continuous improvement rather than sequential, waterfall-style steps. New apps or features may generate operational problems or security vulnerabilities if developers operate independently without taking operations and security into account. These issues or vulnerabilities can be expensive and time-consuming to fix.

Operations

Throughout the delivery and use life cycle of software, operations refer to the methods for controlling its functionality, including: keeping track of system performance, bug fixing, software testing, and mechanisms for software release tuning.

DevOps, which combines basic operational concepts with development cycles, has gained popularity in recent years because these two processes must coexist. Siloed post-development operations can make it easier to find and fix bugs, but this strategy requires engineers to fix software flaws before moving on. This complicates the software's route map.

Organisations can speed up overall efficiency by implementing operations concurrently with software development processes.

Security

All the methods and tools required to create software that is resistant to attack, to swiftly identify and address flaws (or actual breaches), and to construct secure software are together referred to as security.

Application security has traditionally been addressed after development is completed by individuals from the development and operations teams. The development process and response time are slowed considerably by this compartmentalised approach.

Security tools are likewise compartmentalised. Every application security test focuses on that application and its source code. This makes it hard to understand the firm's security vulnerabilities or software risks in the production environment.

Organisations may synchronise the three most crucial steps in the development and delivery of software by integrating application security into a unified DevSecOps process from the beginning to the end.

How Does DevSecOps Differ From a More Traditional DevOps Approach?

Because each stage of the process—design, development, testing, and final approval—is distinct and one stage may only begin after the previous one is finished, traditional software development is frequently referred to as the waterfall technique.

In a majority of firms, the agile methodology, which divides a project into sprints, has essentially supplanted waterfall. However, waterfall-style security testing is still frequent and they are postponed till the end of a sprint! In fixing security issues, this delay causes developers to change directions and go back in time. This "context change" is time-consuming and prone to causing mistakes.

On the other hand, DevSecOps makes it possible for security testing to happen automatically and smoothly throughout the same general period as other development and testing.

To avoid spending time context-switching, developers can, for instance, execute security tests in the development stage in close to real-time. They may also perform security checks in close to real-time throughout production, allowing them to identify any instances of vulnerabilities as soon as they are disclosed.

Why Do You Need DevSecOps?

Over the past few years, the environment of IT infrastructure has changed dramatically. Organisations wanting to thrive and grow through the usage of cutting-edge apps and services have benefited greatly from the change to flexible cloud computing platforms, shared storage and data, and dynamic applications.

Although DevOps applications have advanced significantly in terms of speed, scale, and functionality, they frequently fall short in terms of strong security and compliance. DevSecOps was included in the software development lifecycle to unite development, operations, and security under one roof.

The best techniques to use malware and other exploits are continually being sought after by malicious intruders and hackers. Consider the possibility that malware was inserted into a program during the development process and went undetected until thousands of users gained access to it. Particularly in a world where negative news spreads quickly online, there would be significant harm done to both the customers and the company's reputation.

Any organisation engaged in the creation and distribution of applications must give security the same weight as development and operations. Every developer and network administrator develops and deploys apps with security in mind when DevSecOps and DevOps are integrated.

Implementing DevSecOps in Your Organisation

Implementing DevSecOps is a complex process, as you might anticipate. We'll now look at four steps needed to adopt DevSecOps in your organisation. The following processes are typically present, even though there aren't any obvious, sequential phases that act as a road map.

- Step 01: Planning & Implementation
- Step 02: Development & Testing
- Step 03: Operations & Deployment
- Step 04: Scaling & Monitoring

Step 01: Planning & Implementation

Planning is the key to everything. For successful implementation, the plan must be strategic and short. Simple feature-based summaries won't do. The experts must also build threat models, user designs, and acceptance test requirements.

The next step is development, and teams should begin by assessing the maturity of their current procedures. It makes sense to compile information from several sources to offer direction. A code review system should be established at this point since it promotes uniformity, a feature of DevSecOps.

Step 02: Development & Testing

When it comes to development and building, automated build tools work well. The source code is combined into machine code in such tools via a build script. Tools for building automation include several potent capabilities. They have numerous available UIs in addition to a huge library of plugins. Some libraries are capable of automatically identifying any that are weak and replacing them with new ones.

The pipeline is then put through testing, where a solid automated testing framework instils sound testing procedures.

Step 03: Operations & Deployment

IaC tools are typically used for deployment since they automate the procedure and quicken the distribution of software.

Another critical phase is operation, and operations personnel routinely do periodic maintenance. Zero-day vulnerabilities are terrible. Operation teams should therefore monitor them. DevSecOps can use IaC tools to swiftly and effectively safeguard the organisation's infrastructure while preventing human errors from slipping in.

Stage 04: Scaling & Monitoring

Utilising potent, ongoing monitoring technologies is a crucial component of the process. They guarantee that your security systems are operating according to plan.

Scaling also has a significant impact. With the introduction of virtualization, businesses are no longer forced to squander money on maintaining massive data centres. Instead, they may simply extend the IT infrastructure to handle any dangers that arise.

Some of the fundamental processes in any DevSecOps implementation are listed above. Your project's size and complexity will determine whether your road map includes any unusual extra steps.

Challenges in Implementing DevSecOps

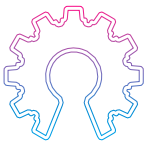
DevSecOps implementation presents some challenges you must overcome.

People and culture

You may need to retrain the members of your DevOps teams, so they are familiar with security best practices and can use your new security tools. Your teams must realise that they are equally responsible for the security of the software they develop and deploy as for its features, function, and usability.

Security tooling

Another problem is finding the right security tools and integrating them into your DevOps workflow. The less training and cultural change are required, the more automated and integrated your DevSecOps tooling is with your CI/CD process.



Open-source software

It's not always the best idea to switch to an automated version of the security tools you've been using for years. Why? Your development environment has probably seen a significant transformation over the last few years. The average modern software programme is made mostly of open-source code. Unfortunately, typical security methods were not created to identify vulnerabilities in open-source software.

Similarily, modern cloud native programmes operate in containers that can swiftly spin up and down. Even the technologies that now call themselves "cloud security" can't figure out how dangerous containerised applications are because they were made for production environments.

Benefits of Adopting DevSecOps

Your product sales can go up if you use DevSecOps. A DevSecOps approach can help you increase your overall security. You may spot vulnerabilities very early on in your workflow, which makes fixing them much simpler. The constant monitoring improves your ability to find threats. The easier it is to sell something commercially, the more secure it is.

When vulnerabilities are found early in the software development cycle, the expense of remediating them can be greatly reduced. Having many teams collaborate on security enhances responsibility. This makes it easier to develop quick and efficient security response plans and more strong security design patterns.

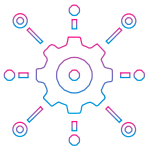
Additionally, DevSecOps reduces the incidence of security bottlenecks. There is no reason to hold off on doing security checks until the development cycle is complete. The delivery of goods is accelerated for these two reasons.

DevSecOps plays a significant role in maintaining compliance with rules set by the industry. One must handle data with particular caution in light of regulations like the General Data Protection Regulation (GDPR). Managers are given a complete overview of these steps through DevSecOps, which **improves the framework for simpler compliance.**

DevSecOps Best Practices

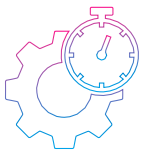
Security must be incorporated into DevOps pipelines for organisations that seek to bring together application developers, security teams, and IT operations. Instead of adding security after development, the goal is to make it an integral part of the software development process from the start.

The best practices listed below will ensure that the DevSecOps process goes smoothly:



1. Automation

Delivering software quickly is a key component of DevOps, therefore adding security shouldn't undermine this goal. You can ensure quick delivery of your applications by integrating automated security controls and tests early in the development cycle.



2. Focusing on Efficiency

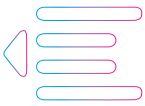
Your workflows are only becoming more secure. You can identify security flaws early by employing technologies that can scan code as you develop it.

3. Threat Modelling

Exercises in threat modelling can help you identify the weak points in your assets and close any security control gaps.

4. Secure Coding

Secure coding creates flaw-resistant software. Without a secure code, a company's confidential information may be hacked. Your developers must have the necessary skills, even if it costs time and money. Coding standards help developers write well-organized code.

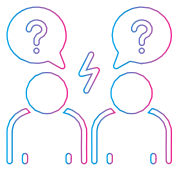


5. Shift Left

Shift-left testing is when you build security into your apps from the outset. This helps you uncover issues and address them sooner. Fixing issues early will be cheaper. It's a great field to study, yet it has issues. Shifting left can hinder your DevOps workflow. Even though it might be hard to solve, using DevSecOps is a good idea in the long run.

Misconceptions About DevSecOps

DevSecOps is no different from other popular jargon in that it has anti-patterns. Let's talk about certain widespread misunderstandings.



1. You need heavily specialised developers

Scripts should have all the components you need to create something from a single command. Web server files, database scripts, and application software are all included in this. The code should be automatically packaged and compiled into a usable application by the CI processes.

2. DevSecOps and Agile are interchangeable

Agile is complemented by DevSecOps, but it does not replace it. For enterprises to get the greatest possible financial benefits, these two must coexist. Agile encourages teamwork and ongoing feedback. Contrary to DevSecOps, it does not cover the delivery of software via QA, testing, and production. By providing approaches and tools to support agile modifications, DevSecOps completes the picture.

3. DevSecOps is a product that can be bought

DevSecOps, such as release management and CI/CD tools, certainly needs to be purchased. However, as DevSecOps is a methodology or a mindset, you cannot purchase the full DevSecOps process. Collaboration across teams and an emphasis on team accountability and ownership are things you can't buy, but they make a huge difference in your organisation.

Considerations When Implementing DevSecOps

DevSecOps engineers are in great demand as more and more businesses come to understand their importance. What will the top performers contribute?

A few additional skill sets are required for the job of a DevSecOps engineer. DevOps ideas, techniques, and culture must be thoroughly understood. Candidates ought to be well-versed in programming languages like Python, Java, and Ruby. A skilled DevSecOps engineer will also be familiar with tools like Chef, Puppet, Checkmarx, and ThreatModeler.

In addition, DevSecOps engineers should also understand the nuances of threat modelling and risk assessment approaches. They also need to be knowledgeable about current cybersecurity dangers, best practices, and other relevant software. DevSecOps experience is ideal in terms of work experience. However, existing expertise in traditional DevOps IT security might serve as a good predictor of future DevSecOps success.

Conclusion

Without a doubt, DevSecOps is redefining the way businesses approach security. However, many mid and low-level firms are still hesitant to adopt DevSecOps for several reasons, including a lack of understanding of what it is, an unwelcome change in employee culture, financial limitations, and perhaps just the vagueness of the name.

The advantages that enterprises can gain from using DevSecOps, both technologically and commercially, are quite exciting. Implementing DevSecOps will benefit your company much in the long term, even though there will undoubtedly be some initial difficulties. This is why working with a reputable solution provider like Codification may help.

Get in touch with Codification

Visit our website to learn more: www.codification.io/services

About Codification

Codification is a Cloud Native transformation consultancy, with a team of over 100 engineers, consultants and business professionals distributed across the world. We were founded in 2019 in the United Kingdom. We have grown since then to have a presence in Europe, the Middle East and Asia, serving leading multinational corporations, government institutions, global banks, and industry giants with our consultancy and expertise.

Through our experience, we have noticed that visionary leaders want to transform their organisations into technology companies to thrive in the new digital-first economy. Here, businesses want to release software faster, improve quality and build a continuous improvement culture where the best ideas win. At Codification, we establish the direction of a company's technological transformation journey and help implement new technologies and processes, resulting in a modernised digital-ready organisation.

CODIFICATION

Codification United Kingdom
The Core
Bath Lane
Newcastle upon Tyne
NE4 5TF

Phone: +44 01670 223994

Web: www.codification.io/