eficode

GUIDE

DevOps for Executives

Foreword

This may come as a surprise: DevOps is no longer a competitive advantage. Today, DevOps adoption is so widespread that it has become an industry standard. The question is no longer: why should my company prioritize a DevOps transformation? Instead, it is: will we survive if we don't?

Successful DevOps adoption eases the pressure off engineering departments and top management. Software projects reach the finishing line faster, simpler, and in the process every department benefits.

Understanding the big picture of DevOps makes choosing solutions and planning a way forward for your company easier. This guide will give you the understanding of DevOps you need to scale it across your organization and improve business outcomes. It will walk you through:

- Why DevOps matters
- Important DevOps practices: cultural and technical
- What a successful DevOps transformation looks like

We hope you have
a pleasant read.

Marko Klemetti

Excel or die

DORA's state of DevOps report showed that one industry tends to perform better than others when it comes to software delivery performance. Namely, the retail industry. The market is so fierce that if your technology organization doesn't excel, you will be surpassed by the competition.

What is Eficode?

Eficode is driving the DevOps movement across seven countries with ideas that put customer value and team satisfaction on center stage. Eficode was doing DevOps before the term even existed by advising global brands on how to make software more effectively. Today, Eficode transforms companies with unmatched DevOps expertise and solutions like the Eficode ROOT DevOps Platform, a DevOps Toolchain as a service. Eficode's community of more than 300 professionals is building the future of software development together.

Contents

Introduction 5

Cultural practices 9

Technical practices 16

Leading successful DevOps transformations 23

Conclusion 29

Introduction

Software delivery drives performance

The Harvard Business Review found that since 2000 over half of the companies listed on the Fortune 500 had gone bust, been bought, or no longer existed. In a world where consumer needs evolve fast and the competition is fierce, business agility and innovation are a must.

DevOps is the cornerstone of business agility. Why? Because DevOps practices drive software delivery performance and software delivery performance drives business outcomes. This is a fact proven by DORA, a DevOps Research and Assessment entity that has surveyed 31,000 software professionals over the past six years.

Validate Ideas

DevOps is an approach to making software that, among other benefits, enables the quick validation of business hypotheses.

It uses automation and state of the art software tools. DevOps breaks down silos between business units to build fast flow of value across the entire organization. DevOps builds a healthy culture, so every team member can do their best work.

DevOps is about more than processes and tools

DevOps helps companies adapt to market needs by delivering valuable software. Naturally, there are a lot of technical concepts involved. However, to think that DevOps is only a matter of technology would be a mistake.

The biggest challenges enterprises face when it comes to DevOps are organizational and cultural. For example, if developers are spending too much time in meetings, automation isn't an easy solution to a lack of productivity. Innovation happens in a culture of sharing, safety and risk-taking. No amount of investment in technology can make up for a culture that is lacking.

Top management can encourage behavior that leads to technical excellence, psychological safety, and continuous improvement. If change management projects aren't driven by the executive team at the very top of the organization they are unlikely to succeed.

DevOps practices

There is no universal solution for implementing DevOps. It can't be simply downloaded and installed. Instead, DevOps transformations need to be culture driven by the leadership, and this guide will explain what that involves. Conceptually, DevOps is often defined as CALMS (Culture, Automation, Lean, Measuring and Sharing).

- **Culture**
- **Automation**
- **Lean**
- **Measuring**
- **Sharing**

Ask your software engineers

"How much time do you have to wait to get a new virtual machine?" If the answer is something other than "no time at all", DevOps presents huge gains for your organization.

Have you looked at my ticket recently?

Yeah, I'm done. I just need the virtual machines I ordered for you.

It's been two months now ...

Works on my machine
¯_( )_/¯

Cultural practices

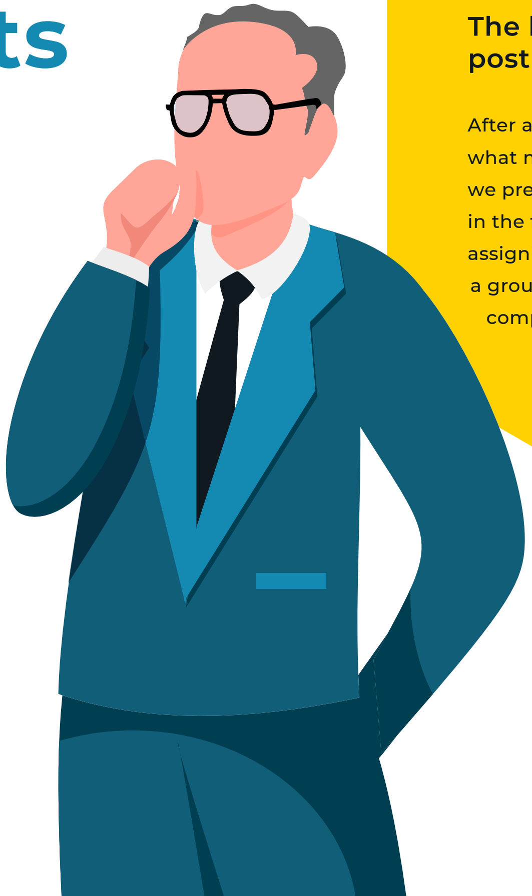
Build trust and prioritize information flow

“Culture eats strategy for breakfast”

PETER DRUCKER
THE FOUNDER OF MODERN MANAGEMENT

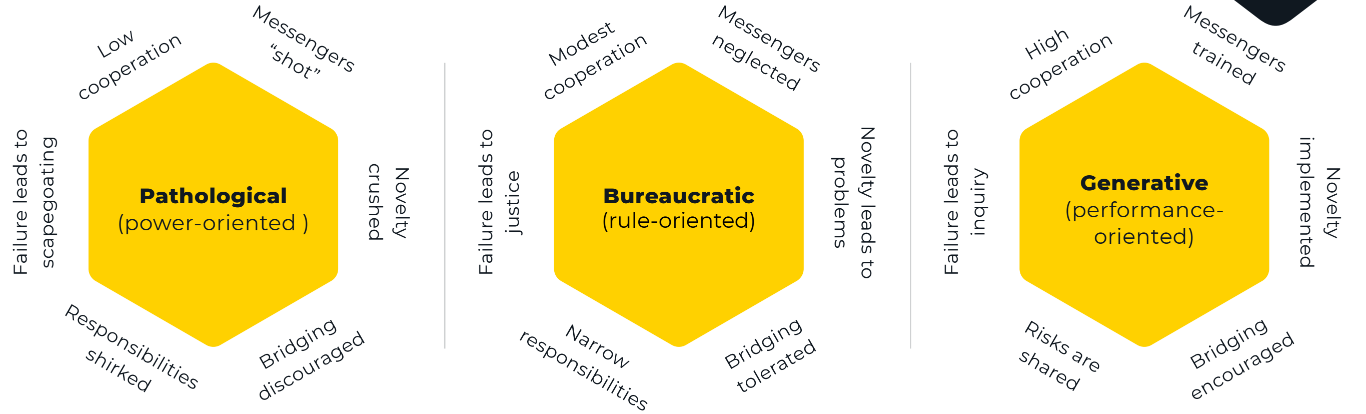
Cultural patterns tend to override strategic efforts, no matter how skillfully crafted the strategy.

A healthy culture is one where there is transparency and clarity. Team members can bring their whole selves to work in a psychologically safe environment.



The blameless postmortem

After an incident apply curiosity to what may have caused it. How do we prevent this from happening in the future? Do this instead of assigning blame to one person or a group of people. This way the company will learn something.



Ron Westrum, creator of the typology of organisation cultures, describes how the culture of an organization is like the personality of an individual. Just as someone's personality shapes their behavior, so too an organization's culture will map out how its people respond to challenges or opportunities.

Leaders define the system in which culture develops. Great leaders play an active role in this, but even without leadership explicitly shaping culture a system will emerge based on:

- **rewards**
- **punishments**
- **resource allocation**
- **symbolic actions**

The Westrum culture typology predicts both software delivery performance and business outcomes. (**State of DevOps report 2018**, by Forsgren et al.) What's more, it's also easily measurable, and you can deduce concrete actions based on survey results. This makes it a powerful tool for driving cultural change in the right direction.

The DevOps Research and Assessment (DORA) report is essential reading every year.

Are you Agile in name only?

In Agile software development autonomous teams are empowered to work independently towards common goals. Depending on your industry, and the size of your company, this may not be easy. After the initial excitement organizations tend to go back to more controlled structures. Is your software really being developed in an Agile way? Finding this out should be a top priority. In our experience senior management often have a false sense of what it means to be truly Agile.

Ask your leadership team...

- Do teams deliver working software to a group of real users every iteration and gather feedback? This includes the first iteration.
- Are teams empowered to change the requirements based on user feedback?
- Are teams empowered to change their process based on what they learn?

Source: The American Department of Defense Innovation Board

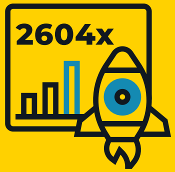
Approval by an external body is not the best way forward

Increasing control and managing risk can decrease innovation and productivity. Did you know that risk can actually decrease when you speed up your processes? In heavily regulated industries in particular, big changes to operations do not lead to speed and quality.

Change approval processes have a tendency to grow complex and confusing over time. This leads to longer feedback loops and longer time to market. Today, thanks to DevOps, change approval boards (CABs) can coach and strategize. The CAB should make sure that the process is transparent and work towards optimizing it to deliver value to customers.

According to the 2019 DORA State of DevOps

- Elite performers had a 7x lower change failure rate than low performers



- Elite performers were 2,604 times faster in recovering from incidents than low performers

- Low performers were 2.6 times more likely to have change management processes that need the approval of an external body

Consolidate tooling with platform teams

Maintaining tickets and requirements in many systems increases more than just the overhead. It also contributes to errors, rework and misalignment. Requirements should bind together business targets and all stakeholders. They also make product development more transparent.

The trick is to connect business metrics to your requirements management workflow. This can be done by consolidating tooling with a platform team, either in house or as a service.

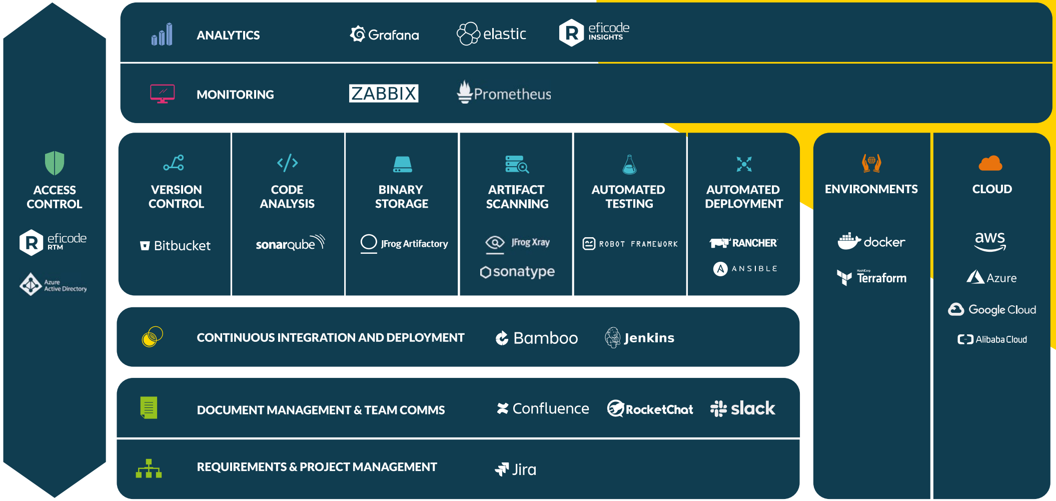
The requirements can then be tracked through each step, all the way to a release. Increasing the level of automation and other benefits of DevOps can be seen in the metrics. Having this upfront traceability also enables auditing and compliance by design.

Requirements management helps everyone see their place in the big picture.

eficode

What is a platform team?

A platform team runs a self-service platform of software and DevOps tools. Product teams use this platform for fast delivery and high quality.



Source: Eficode ROOT Reference Architecture

Data-driven is a state of mind

DevOps makes it easier to connect business metrics to the work that your engineering department is doing. This is important because you cannot improve what you cannot see.

Data is only valuable if it leads to action. Encouraging data-driven decision making at a company can be aided by using the right language.

Here's an example statement: "Make the "buy" button pink to attract more attention and get more sales". Instead, encourage statements like this to be phrased as a hypothesis: "We believe making the "buy" button pink will attract more attention and get more sales". This clarifies that the intention is an experiment: we need to measure, validate, and figure out what we want to keep.

The pink button might drive sales down for some unknown reason, but without a data-driven mindset there won't be a way to know what caused the issue.

eficode

Ask your CIO

- How are we using hypotheses as an organization?

Technical practices

Cloud is a key DevOps capability

A common misconception is that cloud is about where your infrastructure is located. For example: “Our company uses cloud because our machines are on AWS”. This is only half the story.

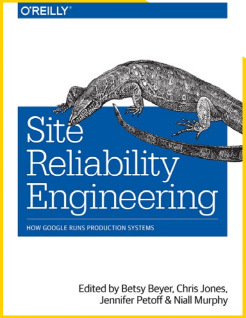
What matters is how you use, manage, and provision your infrastructure.

Simply put, if no changes are made to the way an organization works, where your servers are hosted becomes less relevant. If an engineer has to go through a help desk and wait for days to get a machine to start a new project, the cloud is not being used properly - no matter how big an investment the company has made. To make the most of the cloud you need to build the right architecture and introduce agility across the organization.

Pursue Continuous Delivery in everything you do

Continuous Delivery is a set of technical practices that puts changes of all types into the hands of users safely, quickly and in a sustainable way. Continuous Delivery is about software architecture, automation, and healthy technical practices.

Whether in test, deployment or build, automation is a big part of Continuous Delivery. In the State of DevOps report 2019, it was shown that over 50% of even the low performers had an automated build and automated unit tests. This industry trend could be a wake-up call for those who have not ramped up their automation efforts. In the Site Reliability Engineering book, the engineers at Google define toil as work that is manual, or repetitive, or does not add long-term value. Automation is a way of decreasing toil to make engineers more powerful, better informed, and happier in their work life.


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Murphy

Read now

Automation amplifies the amount of work done. This focus is heavily inspired by the Lean practices from the production industries. Lean is about continuous improvement, optimizing the value stream and minimizing waste. And these practices transfer to the software domain.

Ask your head of engineering

- Can we eliminate rather than automate this process?

Automation is not free or easy to maintain, though. Use COTS (Commercial Off-The-Shelf) tooling such as CMake rather than a set of homegrown scripts to build your software. There is a reason that we use Jenkins, GitLab or Kubernetes. These tools implement the knowledge and best practices of large communities and companies. Use what is available commercially or as open source before building your own solutions.

Build your development community around version control

Managing change in a quickly improving software product can be difficult. What is the difference between two versions? A key ingredient to this traceability is version control.

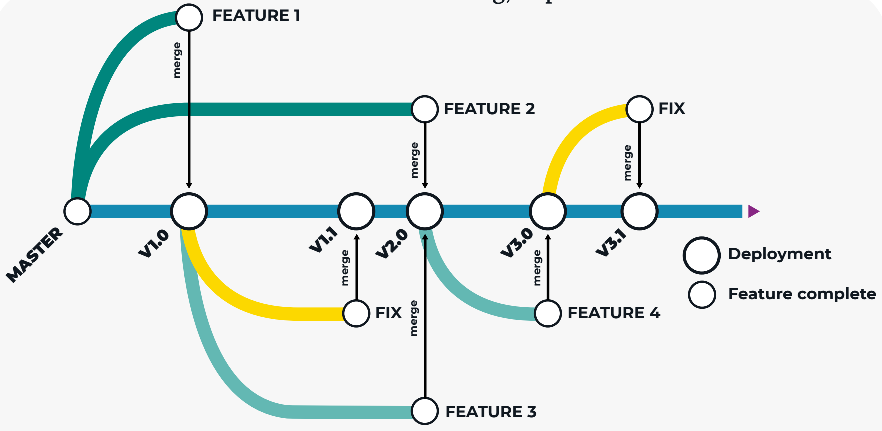
These days the de facto standard for version control is Git and it's the open source tool responsible for managing source code for more than 90% of the world's developers.

Software systems are complex. They're made up of many different elements or deployed in numerous configurations. This means that it is not enough to store only the application source code in version control. You should put the test automation, infrastructure and deployment pipline in there as well.

Version control systems are often connected to task management systems such as Atlassian's Jira Software. This allows for a full trace and audit trail and makes sure that changes happen for a reason. Well-functioning version control also leads the way for introducing new practices like automated security testing, linting, or peer reviews.

Did you know these should also be under version control?

- Configurations
- Declarative infrastructure
- Automated tests
- The way software is built



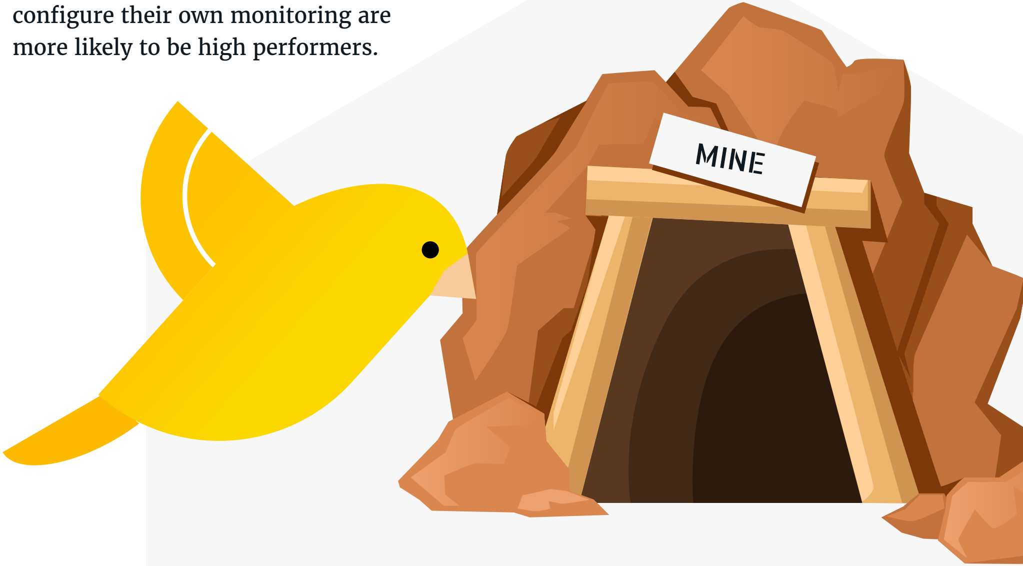
From monitoring to actionable insights

Knowing the current state of the applications you are maintaining enables data-driven decision making. For example, take notice of active bugs in the desktop application, or up-time for an online service.

The aim is to prevent a customer having to notify your engineers that there is a problem. Monitoring can lead to actionable insights and proactivity by:

- Measuring what is believed to be the normal conditions of your applications
- Alerting on deviations
- Becoming aware of alert fatigue
- Using blameless postmortems to avoid recurring incidents

Monitoring and observability should be part of the application design. The Puppet State of DevOps report 2019 shows that teams that are able to configure their own monitoring are more likely to be high performers.



Canary releases are used by Facebook and other big players. You roll out a change to a small subset of users, gather data, and then make it available to everybody.

Quality assurance is more than a gatekeeper

Traditionally, testing and quality assurance have been siloed outside of software teams. In a classic approach to making software, quality assurance (QA) comes in when developers believe they are done. This leads to an adversarial relationship. Developers feel attacked, even though QA is there only to improve the quality of the product.

Companies should place QA in a facilitating role to continuously review the automated tests used by the teams. Are they in a maintainable state? Testers should also help with making test data available and keeping pipelines fast.

There are now tools for efficient automated acceptance testing using natural language. You don't need to know how to code to understand them. An example of this is called behavior-driven development (BDD). Test cases are written so that all stakeholders can understand the purpose of each test. A wider group of people can have discussions about QA that refer to individual test cases.

Any remaining manual testing can be easily automated and included in the Continuous Delivery pipeline.

Database migrations and security testing can all be automated.

Your aim should be that all existing features are tested automatically so that you have a sense of safety before each new release.

Ask your QA team

- Do software teams have what they need to take responsibility for the quality of the product they deliver?

Make security an integral part of your delivery pipeline

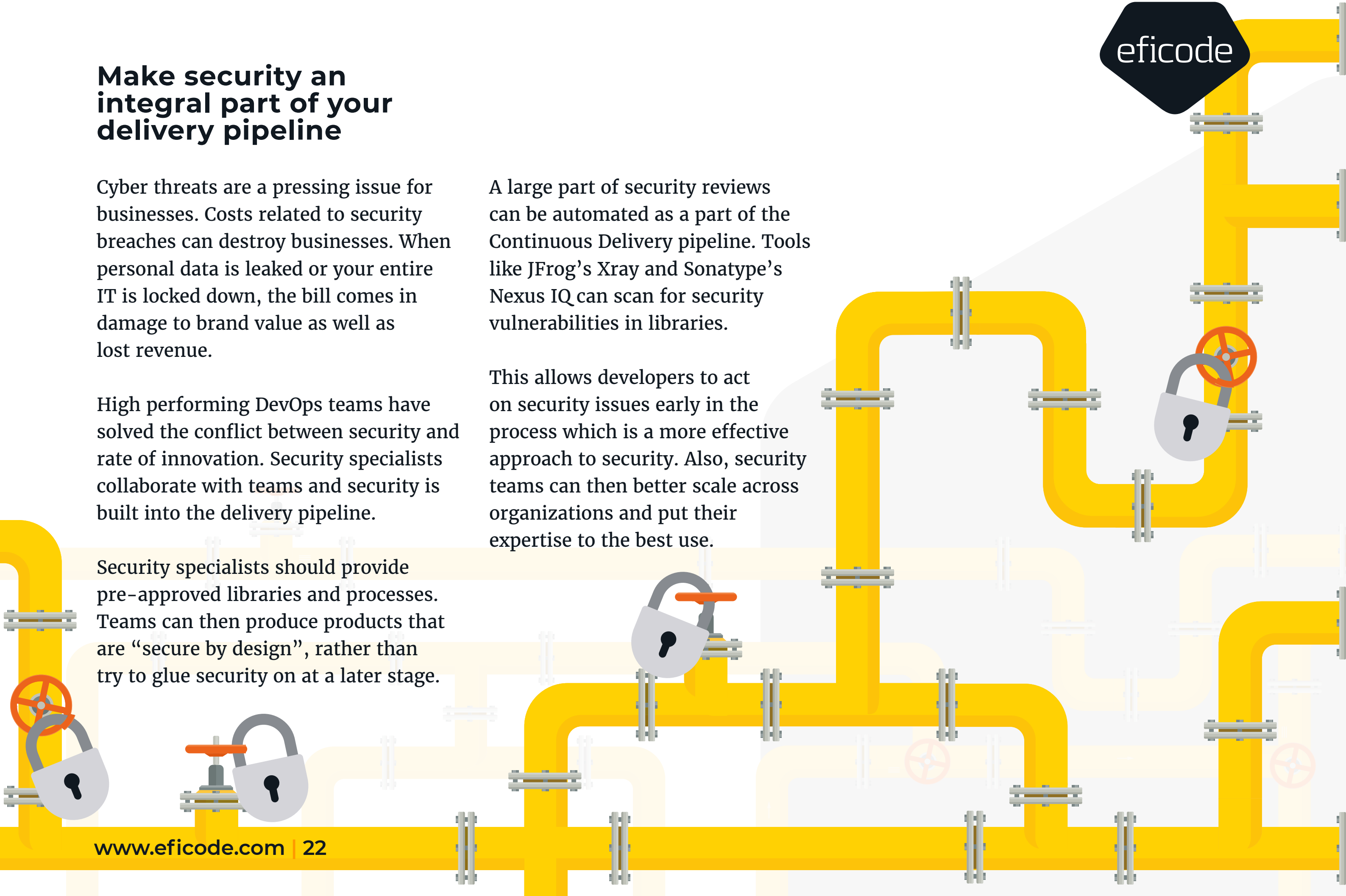
Cyber threats are a pressing issue for businesses. Costs related to security breaches can destroy businesses. When personal data is leaked or your entire IT is locked down, the bill comes in damage to brand value as well as lost revenue.

High performing DevOps teams have solved the conflict between security and rate of innovation. Security specialists collaborate with teams and security is built into the delivery pipeline.

Security specialists should provide pre-approved libraries and processes. Teams can then produce products that are "secure by design", rather than try to glue security on at a later stage.

A large part of security reviews can be automated as a part of the Continuous Delivery pipeline. Tools like JFrog's Xray and Sonatype's Nexus IQ can scan for security vulnerabilities in libraries.

This allows developers to act on security issues early in the process which is a more effective approach to security. Also, security teams can then better scale across organizations and put their expertise to the best use.



Leading successful DevOps transformations

How to scale DevOps

Let's return to the big picture of scaling DevOps at your organization. In Leading Change, John Kotler states that the first part of any change process is to instill a sense of urgency. This is one of the reasons why management buy-in is so important for DevOps transformations. Without enthusiasm from top management, there is a high risk of the organization slipping back into its old ways.

Transform or evolve?

A DevOps transformation changes a company's approach to work. However, it is not a journey that starts at A (very little DevOps) and ends at B (elite DevOps performance).

Elite DevOps performers continue to evolve as part of their company culture and ways of working. At this point, improvements continue organically. However, there may be a period where you need concerted efforts to change ways of working. This is what we call a DevOps transformation. The aim is for this transformation to have an expiration date, so that DevOps becomes part of your day-to-day and evolves according to your customer needs and new advancements in the field.

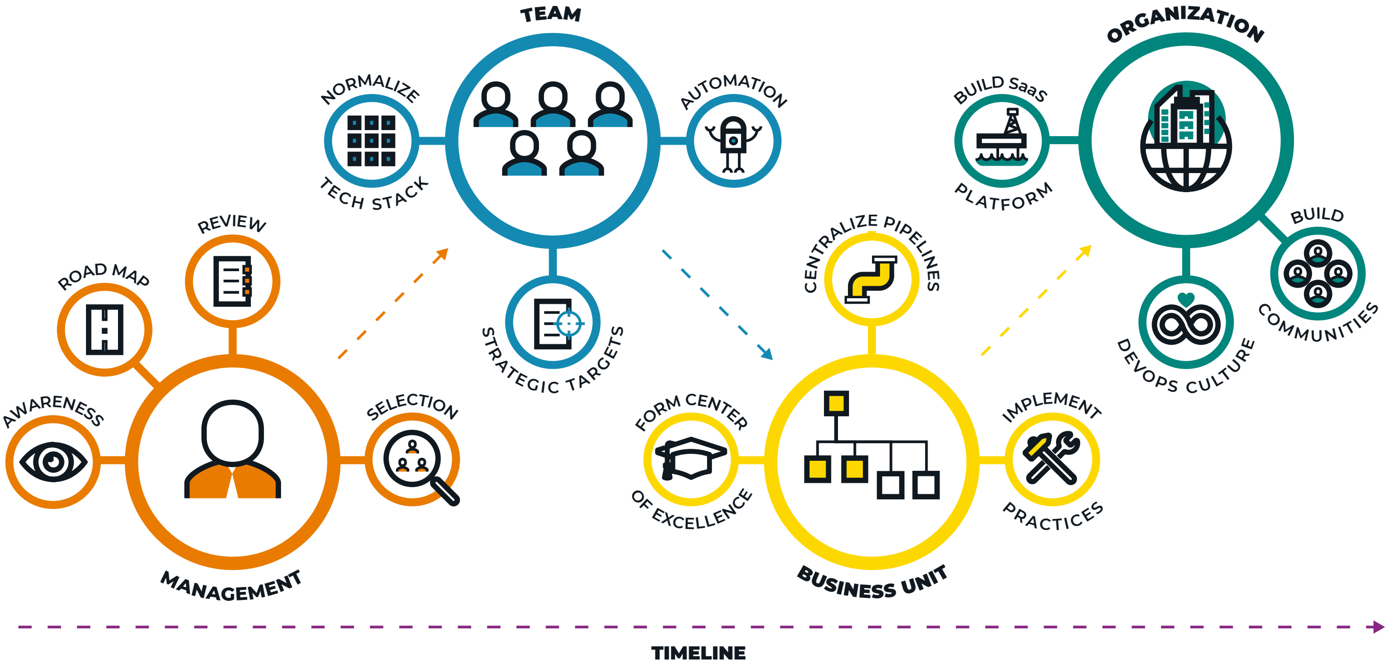
There are many ways to drive a successful DevOps transformation. In fact, there is no "out of the box" solution that applies equally to all organizations.

Ask your leadership team

- Does our DevOps transformation have an expiration date?

The roadmap

DevOps transformations do go through broadly similar phases, though. We've seen the following transformation outline work in practice. For example, each organization will have their own transformation roadmap and their own way of normalizing the tech stack.



A pipeline can provide a safety net to guide you back if you get lost in your transformation. The stages are sequential, meaning it's difficult to skip a stage.

To be able to sustain a DevOps adoption, scale it, and root it in the culture, it's vital to get the basics right. Start by building a healthy culture, with clear change approval processes, basic automation, monitoring and version control.

Let's now shine a spotlight on two aspects of successful DevOps transformations.

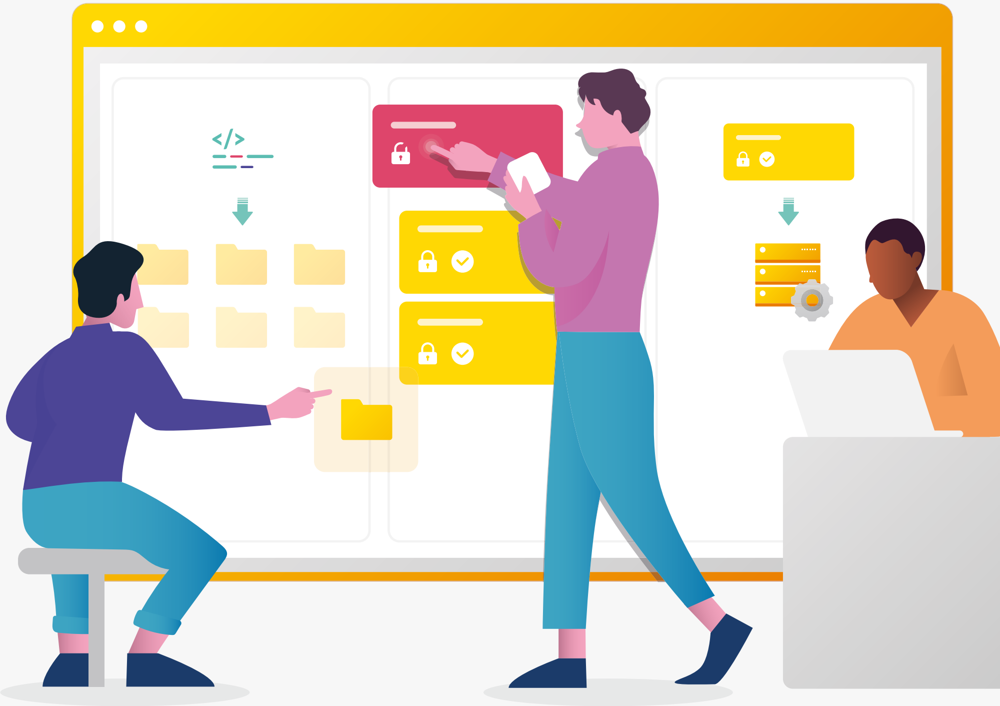
Communities of practice and community builders

What is a community of practice?
A community of practice is a group of professionals with a shared interest in a topic. They actively and voluntarily work on improving their knowledge and skills in that arena together.

Building communities of practice is a healthy way of scaling DevOps. The Communities of Practice are grown into self-sustaining groups. These groups help to keep the organization on track. They also help retain and evolve the technical skills needed inside the organization over time. Building communities increases resilience against reorganizations and product changes.

This is similar to the "guilds" concept in the well-known Spotify organizational model.

Starting and maintaining such communities requires both upfront and continuous investment. Communities



exist organically. They're social entities around a passion for a topic. It's their voluntary nature and dynamism that drives the growth of shared knowledge.

Communities cannot be managed in the way that departments can be. The community itself should define how it wishes to exist.

That being said, there are many different forms of internal leadership that can happen within a community, such as:

- A thought leader
- Someone to organize activities and meetings
- Someone who documents practices

We like to call these internal leaders community builders. They are especially crucial in the beginning if no community of practice exists yet.

You may have a community of practice in your organization without realizing it.

Gather a group of community builders during your DevOps transformation. In more hierarchical organizations this will be known as a Center of Excellence (CoE).

During the transformation, these community builders can create a sense of urgency. They then foster their own communities of practice and take leadership roles within them.

A community of practice for actionable insights

A community of practice for Continuous Delivery

A DevOps transformation Center of Excellence

When DevOps meets design

Digital services take a lot of design. User experience, user flows and branding all need to synchronize with the work of your engineering department.

Just like software, designs evolve. When you think of software as a living entity, it makes sense to have a library that is always kept up to date for designers and developers to refer to. Because your last project is already out of date.

It's a way for design and development to work together more smoothly. When combined with DevOps practices such as automation, they can lead to new products being released very quickly.

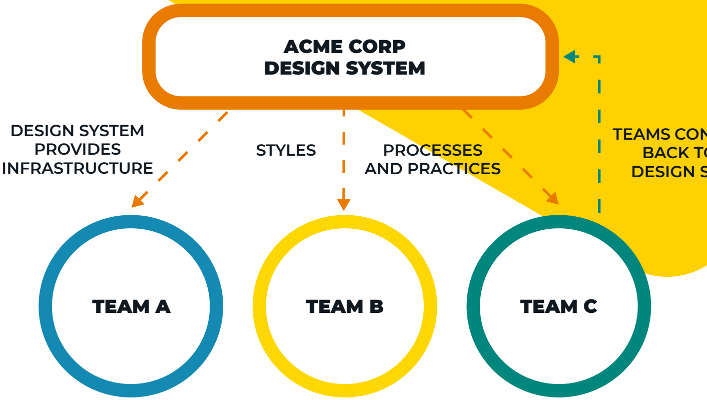
What is a Design System?
Simply put, a Design System is a set of libraries and processes that breaks down software projects into the simplest components. These components can be reused across projects.

The benefits of a Design System

- **Customer experience**
  Consistency in look and feel maintained across products

- **Pleasant collaboration**
  The design system can work as a middle man. Developers can submit pull requests for the brand team to approve

- **Organizational efficiency**
  They save time. Designers and developers don't have to come up with a solution for the same problem multiple times



ACME CORP DESIGN SYSTEM

DESIGN SYSTEM PROVIDES INFRASTRUCTURE

STYLES

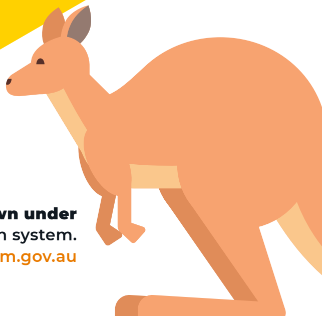
PROCESSES AND PRACTICES

TEAMS CONTRIBUTE BACK TO THE DESIGN SYSTEM

TEAM A

TEAM B

TEAM C



Design down under
The government of Australia has their own design system.
You beauty! designsystem.gov.au

Conclusion

The essential business case for DevOps

As we have seen, DevOps is no longer optional, it is essential. What was once a competitive advantage is now an industry standard and digital transformation is quickly becoming necessary for basic survival. We can prove that DevOps accelerates software delivery and that this in turn drives business performance. For software organizations this is now an "evolve or die" moment.

While technology and tools are a vital aspect of DevOps, it would be a mistake to assume that they offer a complete solution. There is no off-the-shelf product available because DevOps doesn't come in a box. Instead, key stakeholders and decision makers have to take responsibility for driving the necessary cultural changes from the tops of their organizations. A combination of the right tools with the right approach is needed for businesses to realize their full DevOps potential.

Why not start your DevOps journey today?

Book your DevOps Assessment today

Get a clear view of where you are now and how you can reach your goals.

Assessing the current state of your software delivery process, and how satisfied you are with it, is a stepping stone to adopting DevOps.

Get in touch to plan your **DevOps assessment** with our experts.

Contact

If you want to know more, email:
devopsguide@eficode.com

Finland	+358 207 40 11 22
Denmark	+45 31 68 98 75
Norway	+47 48 67 63 60
Sweden	+46 76 340 30 50
Germany	+49 172 4 15 16 17
The Netherlands	+31 20 280 41 18

You can find all our contact information here:
www.eficode.com/about/find-us