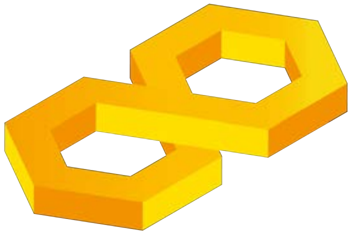
We are starting soon!



Better, faster, stronger with DevOps - but how?

Sofus Albertsen
Academy Headmaster

Linkedin: in/sofusalbertsen/
sofus.albertsen@eficode.com

Building the future of software development

We don't chop wood

We sharpen axes

eficode

78 COUNTRIES

350+ PROFESSIONALS

15 YEARS OF EXISTENCE

Our expertise

DevOps transformation

Set the strategy that will scale DevOps across your organization

Eficode ROOT DevOps platform

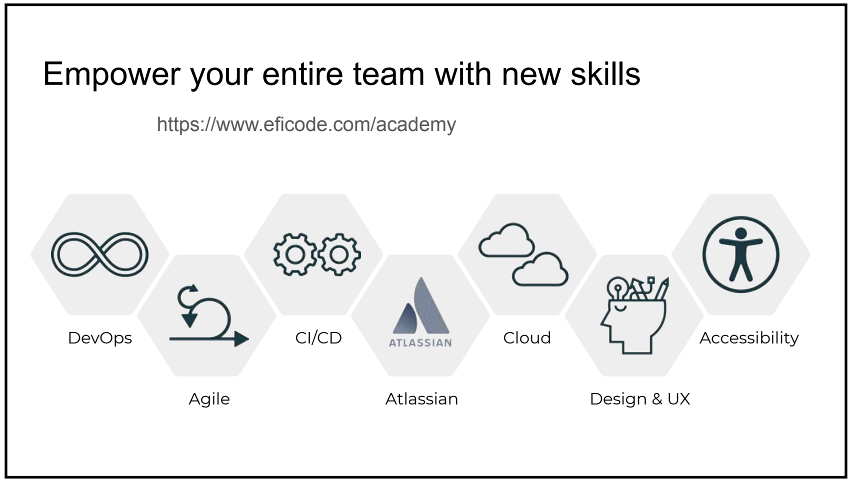
Discover an award-winning one stop shop for all your favorite DevOps tools

Cloud capabilities

Accelerate your digital transformation with the cloud

Digital services development

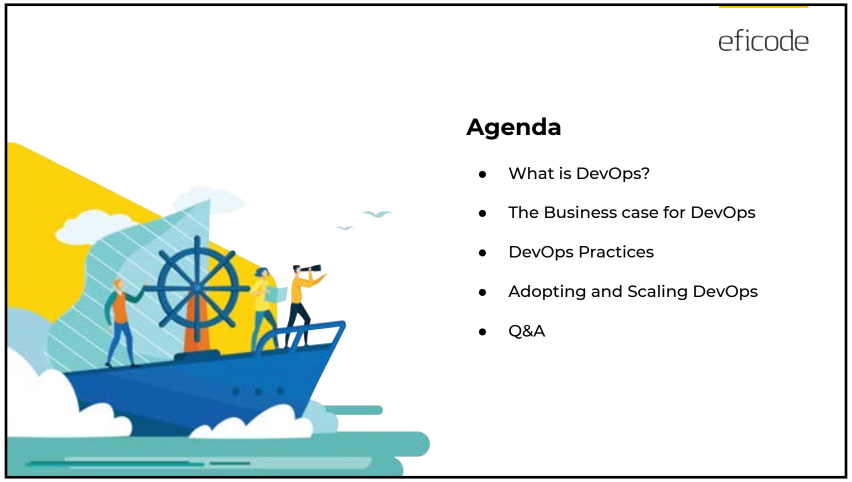
Design and build digital services that create value with agile software, UX and accessibility expertise



Empower your entire team with new skills

https://www.eficode.com/academy

- DevOps
- Agile
- CI/CD
- Atlassian
- Cloud
- Design & UX
- Accessibility



eficode

Agenda

- What is DevOps?
- The Business case for DevOps
- DevOps Practices
- Adopting and Scaling DevOps
- Q&A

Summary

Companies doing DevOps are more profitable

DevOps is easy to explain but hard to do

Culture eats DevOps for Breakfast

Flow and Technical Excellence are key

Menti

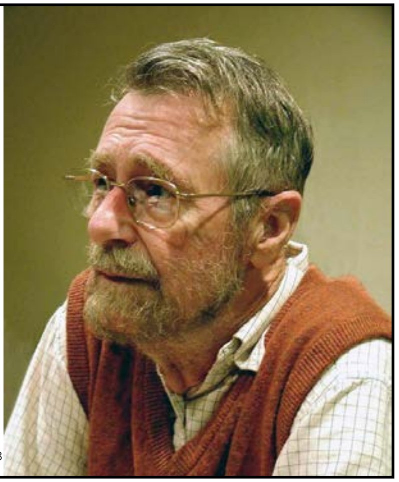
Let's zoom in on you

History

The First Software Crisis

"...as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem."

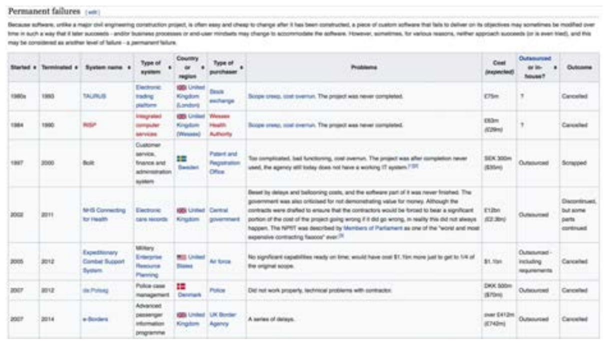
Edsger W. Dijkstra, NATO Software Engineering Conference, 1968



Like building a bridge?

By Kabelleger / David Gubler, CC BY-SA 3.0

Too big?



Permanent failures [edit]

Because software, unlike a major civil engineering construction project, is often easy and cheap to change after it has been constructed, a piece of custom software that fails to deliver on its objectives may sometimes be modified over time in such a way that it later succeeds - and/or business processes or end-user mindsets may change to accommodate the software. However, sometimes, for various reasons, neither approach succeeds (or is even tried), and this may be considered as another level of failure - a permanent failure.

Started	Terminated	System name	Type of system	Country or region	Type of purchaser	Problems	Cost (expected)	Outsourced or in-house?	Outcome
1980s	1993	TAURUS	Electronic trading platform	United Kingdom (London)	Stock exchange	Scope creep, cost overrun. The project was never completed.	£75m	?	Cancelled
1984	1990	RISP	Integrated computer services	United Kingdom (Wessex)	Wessex Health Authority	Scope creep, cost overrun. The project was never completed.	£63m (£29m)	?	Cancelled
1997	2000	Bolit	Customer service, finance and administration system	Sweden	Patent and Registration Office	Too complicated, bad functioning, cost overrun. The project was after completion never used, the agency still today does not have a working IT system.[19][20]	SEK 300m ($35m)	Outsourced	Scrapped
2002	2011	NHS Connecting for Health	Electronic care records	United Kingdom	Central government	Beset by delays and ballooning costs, and the software part of it was never finished. The government was also criticised for not demonstrating value for money. Although the contracts were drafted to ensure that the contractors would be forced to bear a significant portion of the cost of the project going wrong if it did go wrong, in reality this did not always happen. The NPfIT was described by Members of Parliament as one of the "worst and most expensive contracting fiascos" ever.[3]	£12bn (£2.3bn)	Outsourced	Discontinued, but some parts continued
2005	2012	Expeditionary Combat Support System	Military Enterprise Resource Planning	United States	Air force	No significant capabilities ready on time; would have cost $1.1bn more just to get to 1/4 of the original scope.	$1.1bn	Outsourced - including requirements	Cancelled
2007	2012	de Polsag	Police case management	Denmark	Police	Did not work properly, technical problems with contractor.	DKK 500m ($70m)	Outsourced	Cancelled
2007	2014	e-Borders	Advanced passenger information programme	United Kingdom	UK Border Agency	A series of delays.	over £412m (£742m)	Outsourced	Cancelled

Software is not like building a bridge

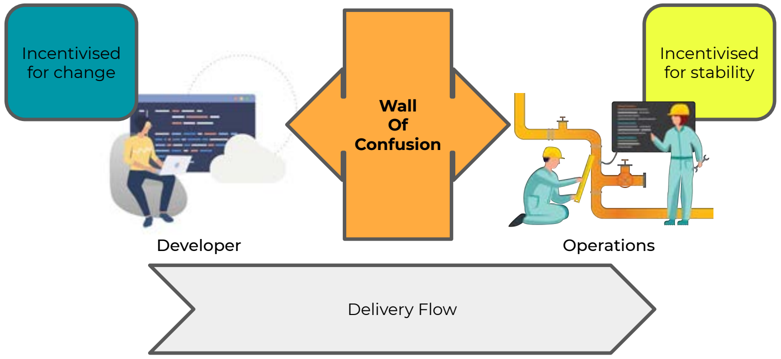
- Well defined requirements
- Same function throughout its lifetime
- Static surroundings
- Well known materials

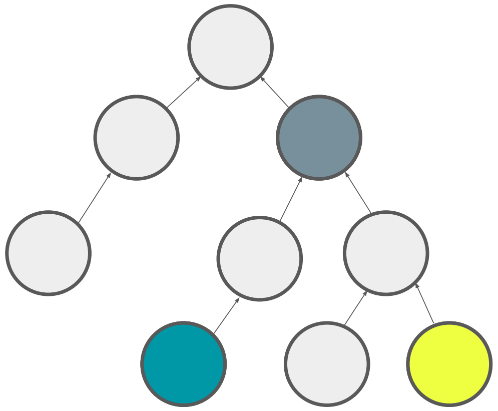
eficode

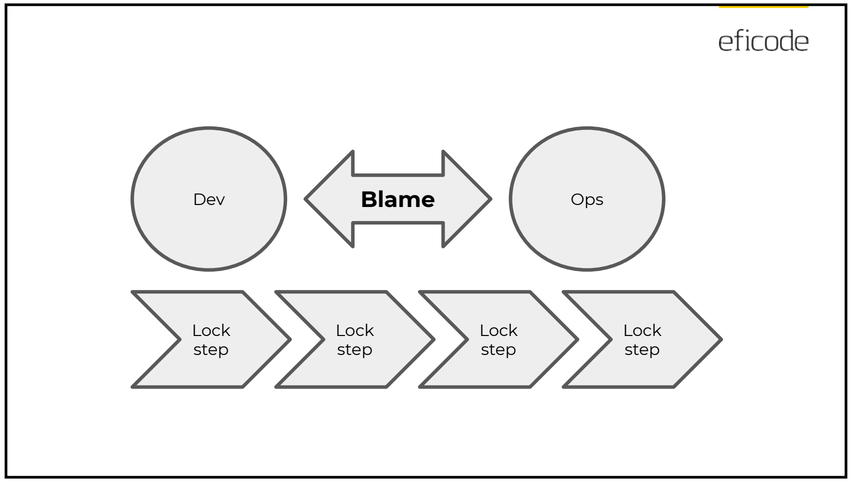
DevOps

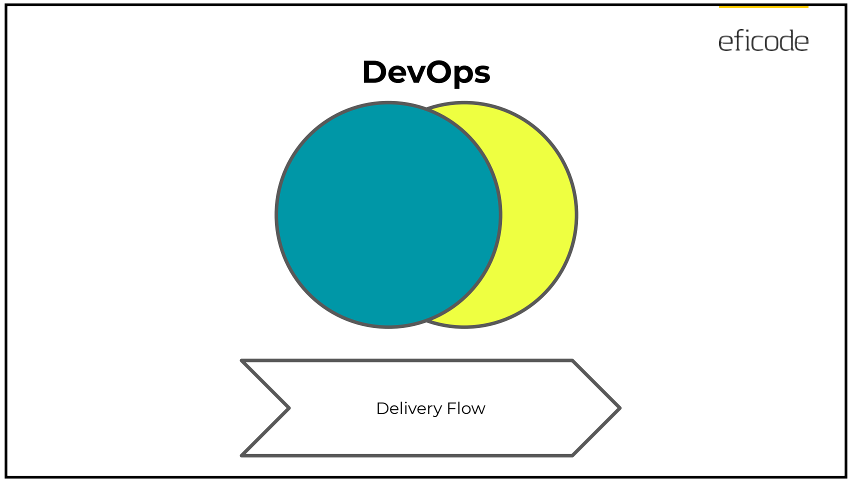
The lightning introduction

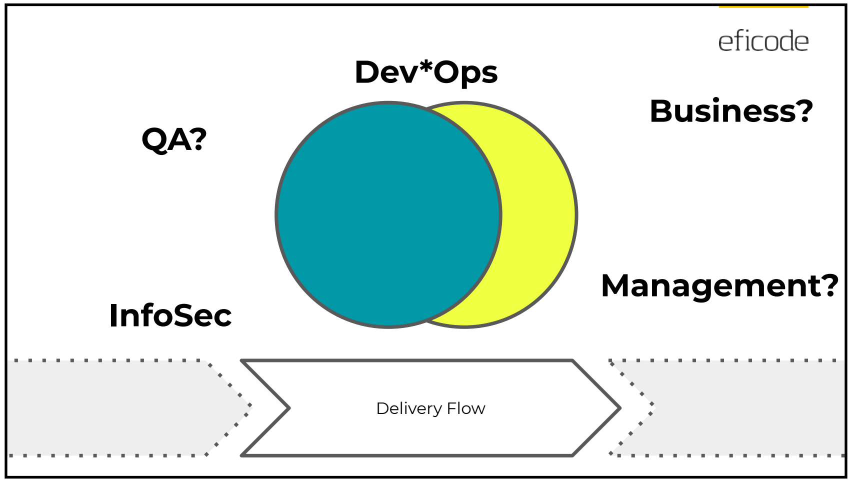
The Chronic Conflict






eficode
Dev
Blame
Ops
Lock step
Lock step
Lock step
Lock step


eficode
DevOps
Delivery Flow



eficode

Dev*Ops

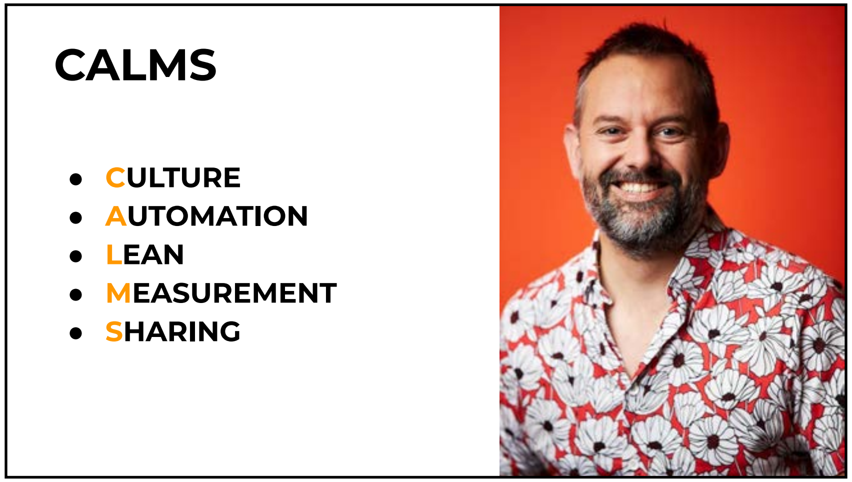
QA?

Business?

InfoSec

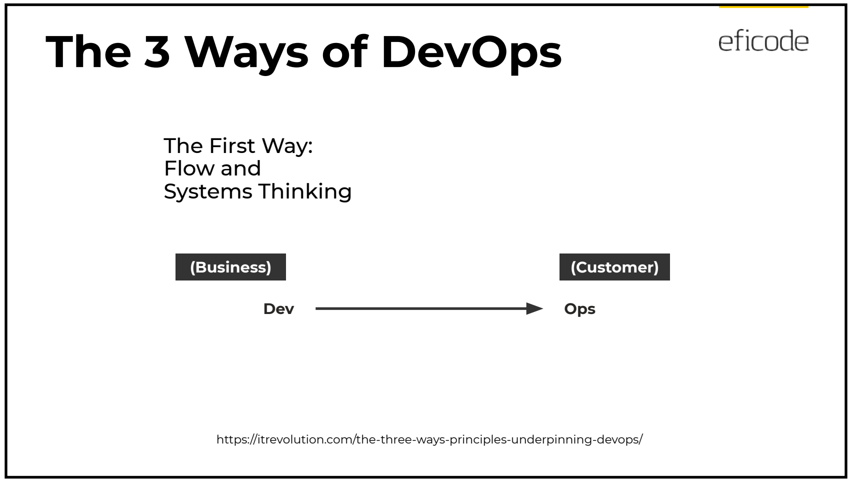
Management?

Delivery Flow



CALMS

- CULTURE
- AUTOMATION
- LEAN
- MEASUREMENT
- SHARING



The 3 Ways of DevOps

eficode

The First Way:
Flow and
Systems Thinking

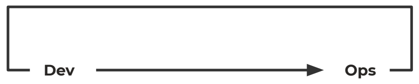
(Business)		(Customer)
Dev	$\longrightarrow$	Ops

https://itrevolution.com/the-three-ways-principles-underpinning-devops/

The 3 Ways of DevOps

eficode

The Second Way:
Amplify Feedback Loops

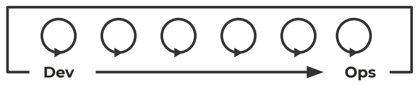


https://itrevolution.com/the-three-ways-principles-underpinning-devops/

The 3 Ways of DevOps

eficode

The Third Way:
Culture of Continual Experimentation and Learning



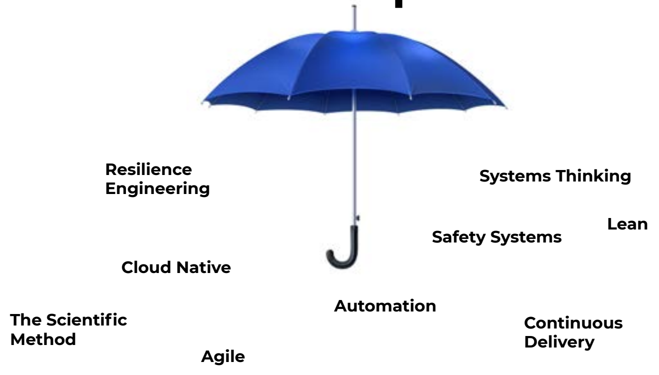
https://itrevolution.com/the-three-ways-principles-underpinning-devops/

Feedback loops

Photo by Christian Bowen on Unsplash

DevOps

eficode



eficode

DevOps

The Business Case

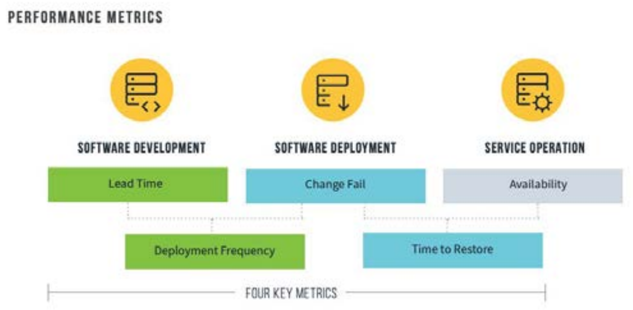
If we get good at DevOps, do we get more money?

eficode

Learn from the research

eficode

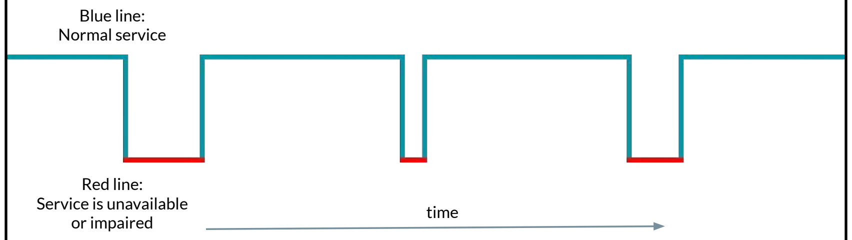
How we measure DevOps



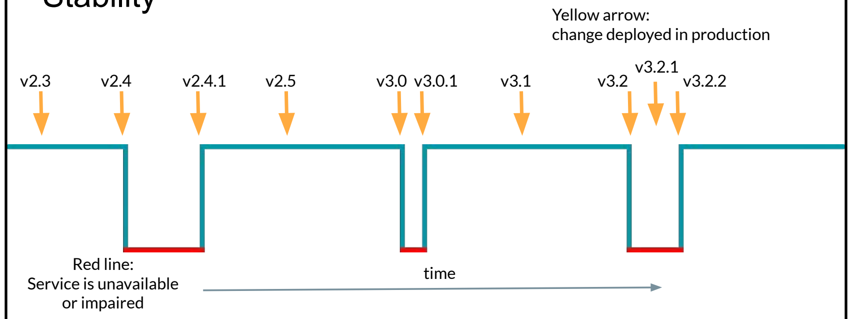
Excerpt from page 16, State of DevOps 2019, Forsgren et al

Availability

Availability is measured as % time the system is available with normal service



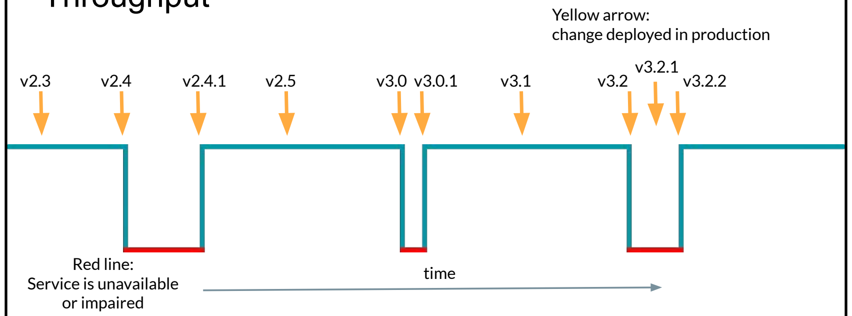
Stability



Stability is measured with two metrics:

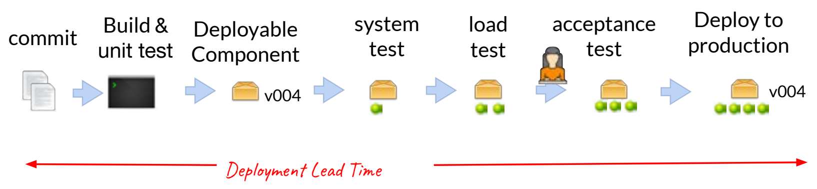
- time to recover
- percent failure rate

Throughput

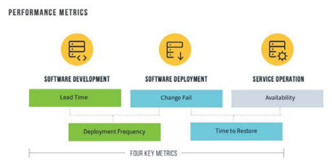


Throughput is measured with two metrics:
- Release frequency
- Lead time from commit to deploy

Deployment Lead Time

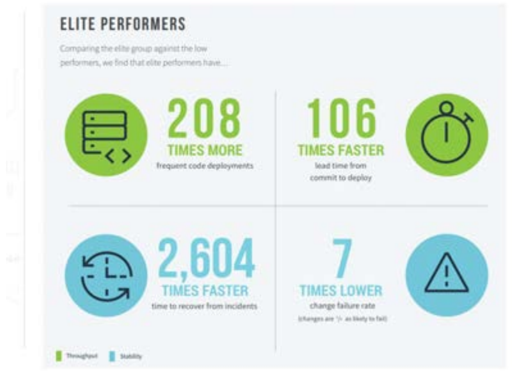


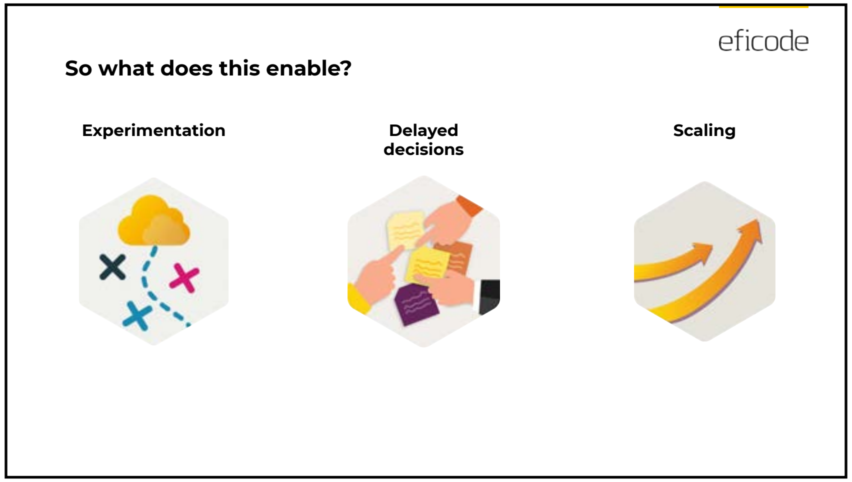
eficode



Ability to meet or exceed organizational goals

How do you score?





eficode

So what does this enable?

Experimentation

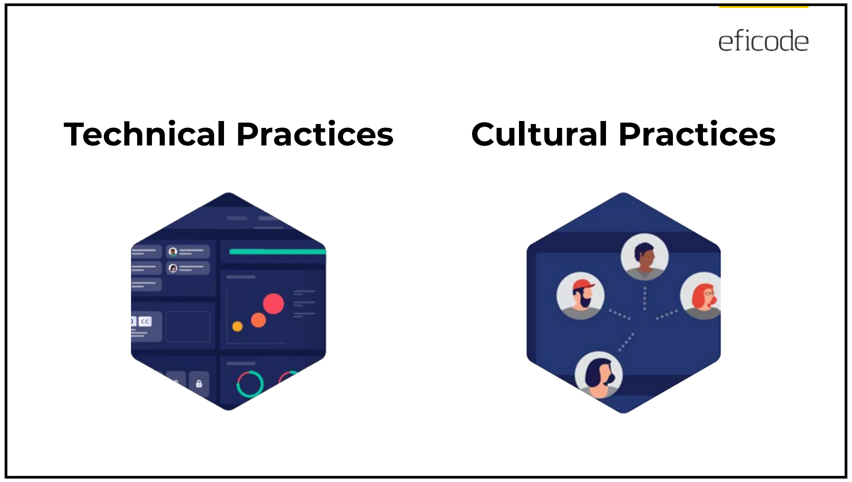
Delayed decisions

Scaling

eficode

DevOps Practices

What do high performers do differently?



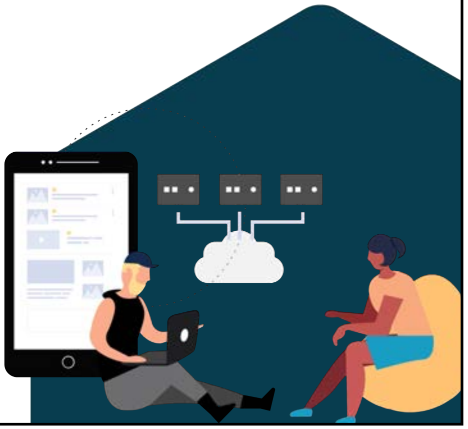
Technical Practices

Cultural Practices

Technical Practices

eficode

Cloud



eficode

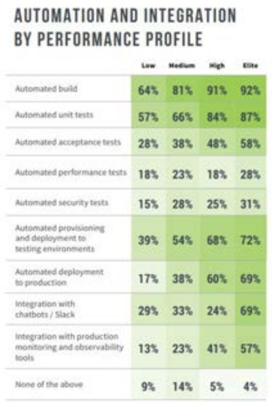
Cloud characteristics

On-demand self-service	Broad network access	Resource Pooling	Rapid Elasticity	Measured Service

eficode

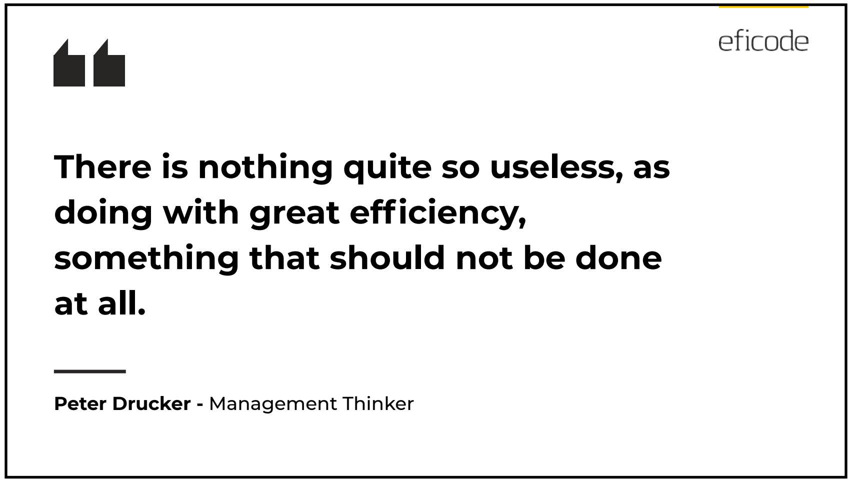
Automation

What do we automate?



AUTOMATION AND INTEGRATION BY PERFORMANCE PROFILE

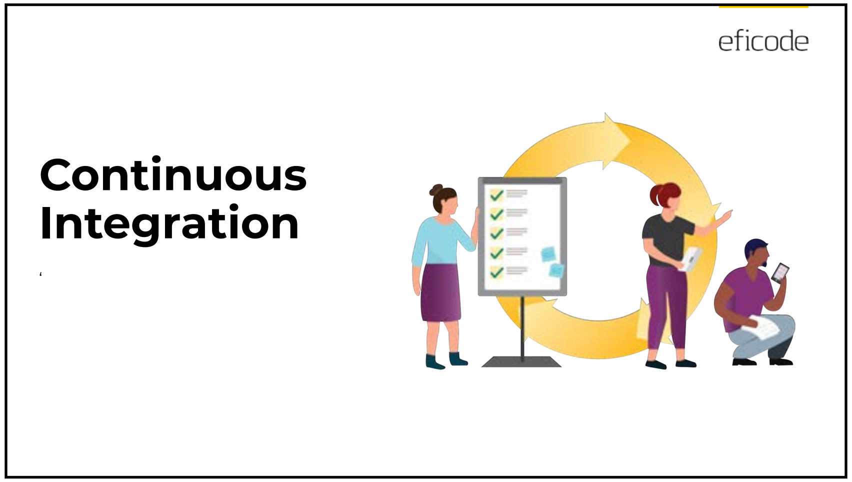
	Low	Medium	High	Elite
Automated build	64%	81%	91%	92%
Automated unit tests	57%	66%	84%	87%
Automated acceptance tests	28%	38%	48%	58%
Automated performance tests	18%	23%	18%	28%
Automated security tests	15%	28%	25%	31%
Automated provisioning and deployment to testing environments	39%	54%	68%	72%
Automated deployment to production	17%	38%	60%	69%
Integration with chatbots / Slack	29%	33%	24%	69%
Integration with production monitoring and observability tools	13%	23%	41%	57%
None of the above	9%	14%	5%	4%



eficode

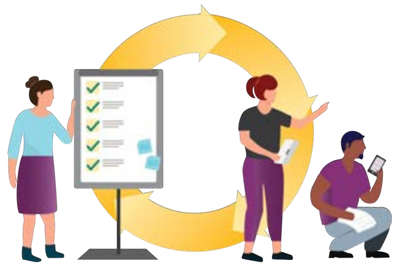
There is nothing quite so useless, as doing with great efficiency, something that should not be done at all.

Peter Drucker - Management Thinker



eficode

Continuous Integration



eficode

Integrate every day

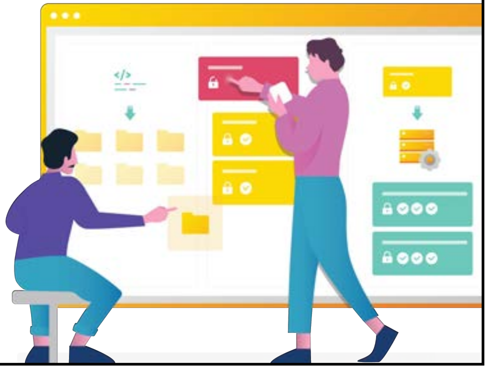
Build every commit

Fix broken builds immediately

eficode

Loosely Coupled Architecture

The team can test, deploy and change their system



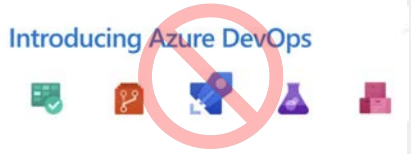
“

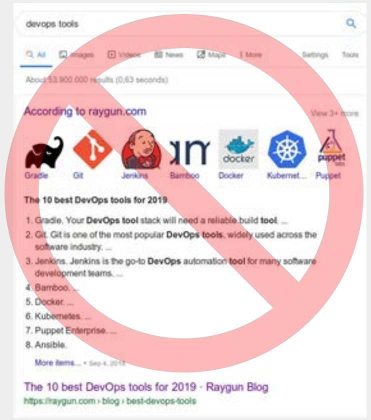
Industry and technology stack, doesn’t matter. Architecture does.

Nicole Forsgren, Ph.d

DevOps tools

Devops is not **a** tool





About 53.900.000 results (0,63 seconds)

According to raygun.com

View 3+ more

The 10 best DevOps tools for 2019

1. Gradle. Your **DevOps tool** stack will need a reliable build **tool**. ...
2. Git. Git is one of the most popular **DevOps tools**, widely used across the software industry. ...
3. Jenkins. Jenkins is the go-to **DevOps** automation **tool** for many software development teams. ...
4. Bamboo. ...
5. Docker. ...
6. Kubernetes. ...
7. Puppet Enterprise. ...
8. Ansible.

More items... • Sep 4, 2018

The 10 best DevOps tools for 2019 · Raygun Blog
https://raygun.com › blog › best-devops-tools

Capabilities of DevOps tools

- **Version controlled code** (Git)
- (Automatic) **Testable code**
- **Infrastructure as code** (Ansible, Terraform, Kubernetes)
- **Runtime and dependencies as code** (Docker, Cri-O)
- **Release pipeline** (Jenkins, CircleCI, Octodeploy)
- **Unified logging and Metrics** (ELK, Prometheus, Greylog)

Cultural Practices

eficode

Decoupled Teams

eficode

Any organization that designs a system will produce a design whose structure is a copy of the organization's communication structure

Conway’s law

eficode

if we have managers deciding . . . which services will be built, by which teams, we implicitly have managers deciding on the system architecture

Ruth Malan, Software Architecture Consultant, Bredemeyer

eficode

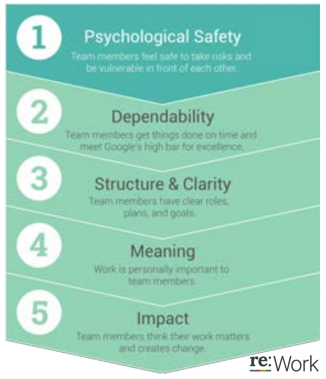
Psychological Safety

menti.com

Let us talk culture

Team performance @ Google

eficode



1. Psychological Safety
Team members feel safe to take risks and be vulnerable in front of each other.

2. Dependability
Team members get things done on time and meet Google's high bar for excellence.

3. Structure & Clarity
Team members have clear roles, plans, and goals.

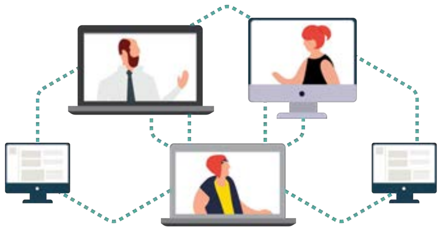
4. Meaning
Work is personally important to team members.

5. Impact
Team members think their work matters and creates change.

re:Work

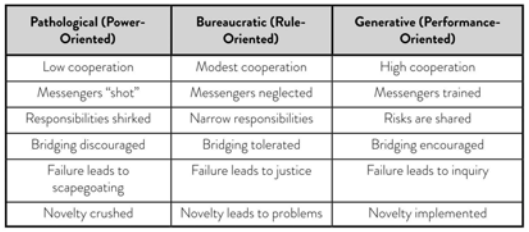
Westrum Typology of Organizational Cultures

eficode



Westrum Typology of Organizational Cultures

eficode



Pathological (Power-Oriented)	Bureaucratic (Rule-Oriented)	Generative (Performance-Oriented)
Low cooperation	Modest cooperation	High cooperation
Messengers "shot"	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

Westrum culture models, Table 3.1 Accelerate

eficode

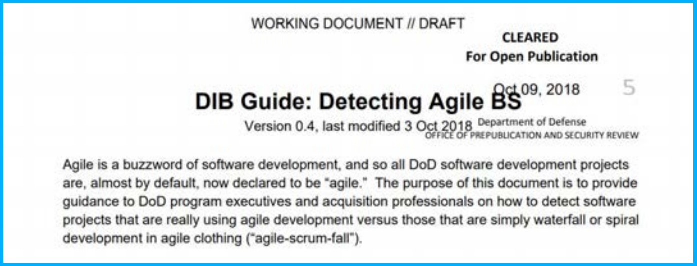
Gitlab nightmare

> We accidentally deleted production data and might have to restore from backup. Google Doc with live notes https://t.co/EVRbHzYlk8
>
> — GitLab.com Status (@gitlabstatus) February 1, 2017

eficode

Transparency

Recommendations from the Defense



WORKING DOCUMENT // DRAFT

CLEARED
For Open Publication

Oct 09, 2018

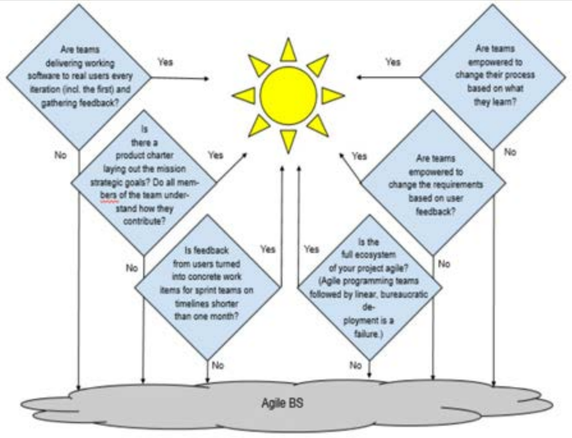
Department of Defense
OFFICE OF PREPUBLICATION AND SECURITY REVIEW

5

DIB Guide: Detecting Agile BS

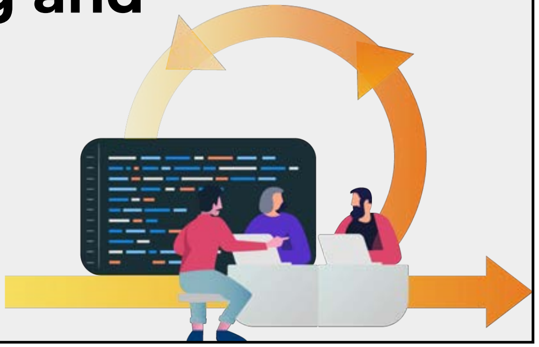
Version 0.4, last modified 3 Oct 2018

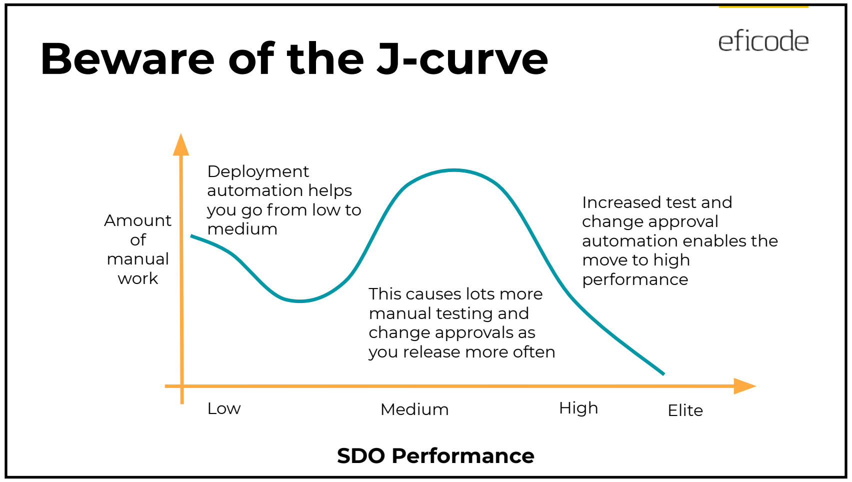
Agile is a buzzword of software development, and so all DoD software development projects are, almost by default, now declared to be "agile." The purpose of this document is to provide guidance to DoD program executives and acquisition professionals on how to detect software projects that are really using agile development versus those that are simply waterfall or spiral development in agile clothing ("agile-scrum-fall").


Are teams delivering working software to real users every iteration (incl. the first) and gathering feedback?
Yes
No
Is there a product charter laying out the mission strategic goals? Do all members of the team understand how they contribute?
Yes
No
Is feedback from users turned into concrete work items for sprint teams on timelines shorter than one month?
Yes
No
Are teams empowered to change their process based on what they learn?
Yes
No
Are teams empowered to change the requirements based on user feedback?
Yes
No
Is the full ecosystem of your project agile? (Agile programming teams followed by linear, bureaucratic de-ployment is a failure.)
Yes
No
Agile BS

eficode

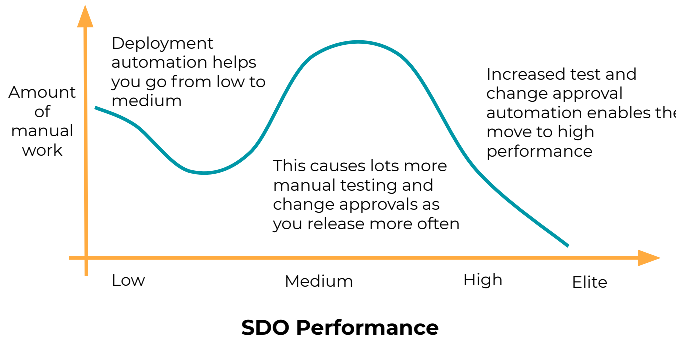
Adopting and Scaling DevOps





eficode

Beware of the J-curve



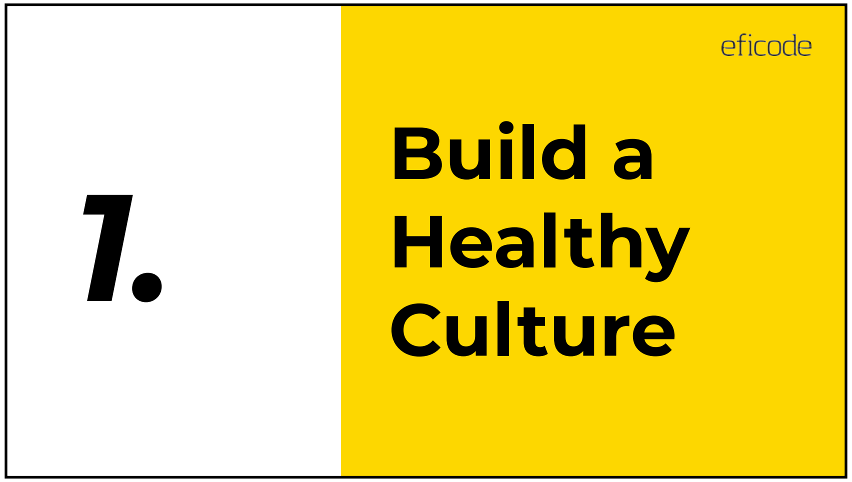
You are special

It just does not matter

James Grenning, Co-Author of the Agile Manifesto and Author of TDD for Embedded C

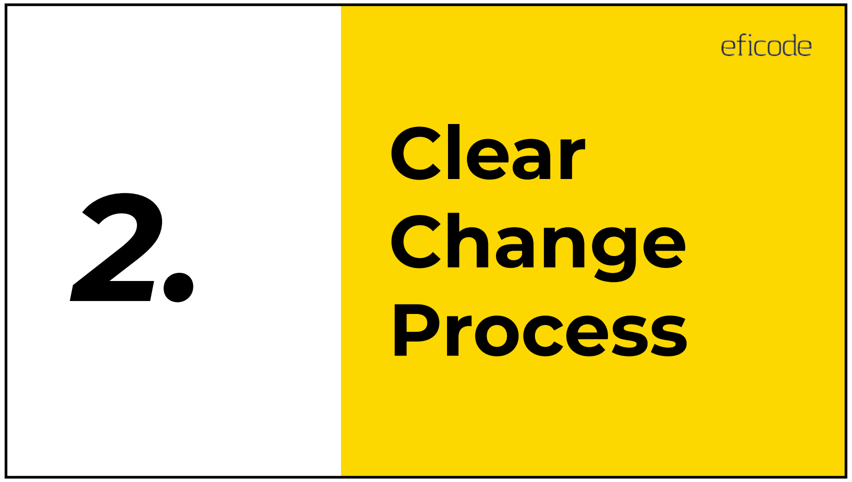
eficode

Summary



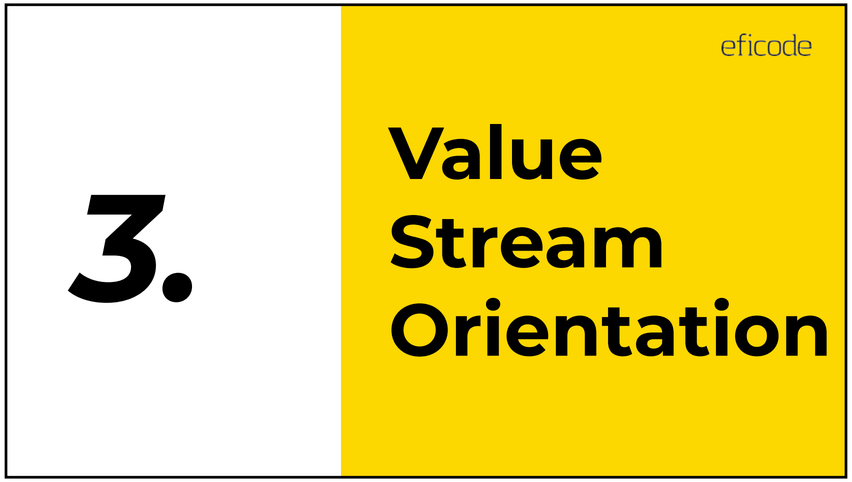
1.

Build a Healthy Culture



2.

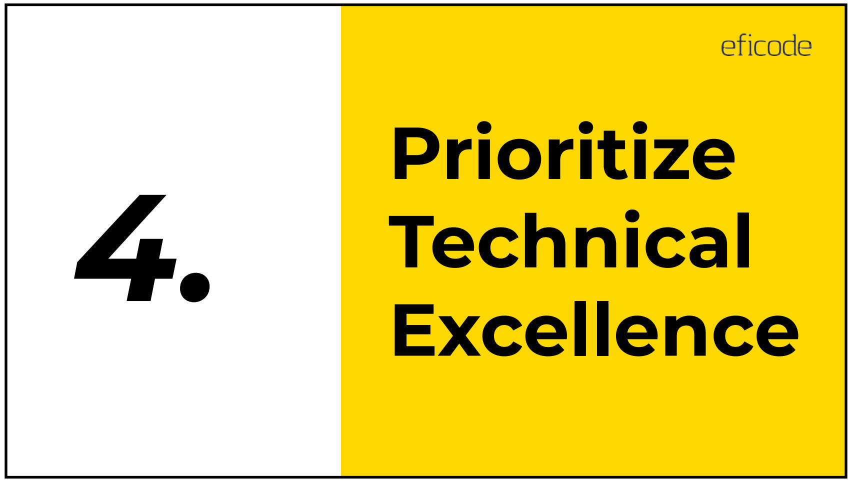
Clear Change Process



eficode

3.

Value Stream Orientation



eficode

4.

Prioritize Technical Excellence

eficode

1. **Healthy Culture**
2. **Clear Change Process**
3. **Value Stream orientation**
4. **Technical Excellence**

eficode

Questions?

Google re:Work

re:Work

Let's make work better.

re:Work is a collection of practices, research, and ideas from Google and others to help you put people first.

Learn more about re:Work with Google.

Where to go from here

Culture

The DevOps Handbook

How to Create World-Class Agility, Reliability, & Security in Technology Organizations

Increase profitability, elevate work culture, and exceed productivity goals through DevOps practices with this non-fiction follow-up to the bestselling *The Phoenix Project*

by Gene Kim, Jez Humble, Patrick Debois, and John Willis

Where to go from here

Technology

The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations

eficode

Thank you!

Sofus Albertsen

sofus.albertsen@eficode.com

Linkedin: in/sofusalbertsen/

eficode.com