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

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

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

Let's zoom in on you
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?
Software is not like building a bridge

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

Incentivised for change

Wall Of Confusion

Incentivised for stability

Developer

Operations

Delivery Flow
Delivery Flow

QA?
InfoSec

Dev*Ops
Business?

Management?

CALMS
- CULTURE
- AUTOMATION
- LEAN
- MEASUREMENT
- SHARING
The 3 Ways of DevOps

The First Way:
Flow and Systems Thinking

(Business)  (Customer)

Dev    -->  Ops

https://itrevolution.com/the-three-ways-principles-underpinning-devops/
The 3 Ways of DevOps

The Second Way:
Amplify Feedback Loops

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

The Third Way:
Culture of Continual Experimentation and Learning

https://itrevolution.com/the-three-ways-principles-underpinning-devops/
Feedback loops
If we get good at DevOps, do we get more money?
Learn from the research

How we measure DevOps

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

Blue line: Normal service
Red line: Service is unavailable or impaired

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

Yellow arrow: change deployed in production

v2.3 v2.4 v2.4.1 v2.5 v3.0 v3.0.1 v3.1 v3.2 v3.2.1 v3.2.2
Throughput

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

Deployment Lead Time

Red line: Service is unavailable or impaired

Yellow arrow: Change deployed in production
Ability to meet or exceed organizational goals

How do you score?
So what does this enable?

- Experimentation
- Delayed decisions
- Scaling

#### DevOps Practices

What do high performers do differently?
Technical Practices  Cultural Practices

Technical Practices
Cloud characteristics:

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

What do we automate?

<table>
<thead>
<tr>
<th>Automation and Integration by Performance Profile</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Elite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automated build</td>
<td>64%</td>
<td>81%</td>
<td>91%</td>
<td>92%</td>
</tr>
<tr>
<td>Automated unit tests</td>
<td>57%</td>
<td>66%</td>
<td>84%</td>
<td>87%</td>
</tr>
<tr>
<td>Automated acceptance tests</td>
<td>28%</td>
<td>38%</td>
<td>48%</td>
<td>58%</td>
</tr>
<tr>
<td>Automated performance tests</td>
<td>18%</td>
<td>22%</td>
<td>18%</td>
<td>28%</td>
</tr>
<tr>
<td>Automated security tests</td>
<td>15%</td>
<td>28%</td>
<td>25%</td>
<td>31%</td>
</tr>
<tr>
<td>Automated provisioning and deployment to testing environments</td>
<td>20%</td>
<td>34%</td>
<td>68%</td>
<td>72%</td>
</tr>
<tr>
<td>Automated deployment to production</td>
<td>17%</td>
<td>38%</td>
<td>60%</td>
<td>69%</td>
</tr>
<tr>
<td>Integration with Chef / Ansible</td>
<td>29%</td>
<td>33%</td>
<td>34%</td>
<td>69%</td>
</tr>
<tr>
<td>Integration with Prometheus and observability tools</td>
<td>13%</td>
<td>23%</td>
<td>41%</td>
<td>67%</td>
</tr>
<tr>
<td>None of the above</td>
<td>9%</td>
<td>14%</td>
<td>5%</td>
<td>4%</td>
</tr>
</tbody>
</table>
“There is nothing quite so useless, as doing with great efficiency, something that should not be done at all.”

Peter Drucker - Management Thinker
Integrate every day  Build every commit  Fix broken builds immediately

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

Introducing Azure DevOps
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)
Decoupled Teams

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

Conway's law
"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

Psychological Safety
Let us talk culture

Team performance @ Google

1. Psychological Safety
   - Team members feel safe to share ideas and be vulnerable to find solutions.

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 know their work matters and impact change.
Westrum Typology of Organizational Cultures

<table>
<thead>
<tr>
<th></th>
<th>Pathological (Power-Oriented)</th>
<th>Bureaucratic (Role-Oriented)</th>
<th>Generative (Performance-Oriented)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low cooperation</td>
<td>Low cooperation</td>
<td>High cooperation</td>
<td></td>
</tr>
<tr>
<td>Messengers &quot;shot&quot;</td>
<td>Messengers neglected</td>
<td>Messengers trained</td>
<td></td>
</tr>
<tr>
<td>Responsibilities</td>
<td>Responsibilities shared</td>
<td>Responsibilities shared</td>
<td></td>
</tr>
<tr>
<td>Bridging</td>
<td>Bridging encouraged</td>
<td>Bridging encouraged</td>
<td></td>
</tr>
<tr>
<td>Failure</td>
<td>Failure leads to scapegoating</td>
<td>Failure leads to justice</td>
<td>Failure leads to inquiry</td>
</tr>
<tr>
<td>Novelty</td>
<td>Novelty leads to problems</td>
<td>Novelty implemented</td>
<td></td>
</tr>
</tbody>
</table>

Westrum culture models, Table 3.1 Accelerate
Gitlab nightmare

We accidentally deleted production data and might have to restore from backup. Google Doc with live notes [https://t.co/EVRbHzYlk8](https://t.co/EVRbHzYlk8)

— GitLab.com Status (@gitlabstatus) February 1, 2017
Recommendations from the Defense
Adopting and Scaling DevOps

Beware of the J-curve

Deployment automation helps you go from low to medium

Increased test and change approval automation enables the move to high performance

Amount of manual work

This causes lots more manual testing and change approvals as you release more often

Low  Medium  High  Elite

SDO Performance
You are special

It just does not matter

James Grenning, Co-Author of the Agile Manifesto and Author of TDD for Embedded C
1. Build a Healthy Culture

2. Clear Change Process
3. Value Stream Orientation

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

Questions?
Thank you!

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

eficode.com