

# Changing Culture and Building Confidence

*An excerpt from **Lessons from 29 DevOps Experts On The Best Way to Make The Transition to Continuous Delivery***



# CHANGING CULTURE AND BUILDING CONFIDENCE

**C**ontinuous delivery means making sure that the software you produce is always ready for release. That much is clear. What may be less obvious is that to get there, an organization must be ever mindful that something else besides production continuously evolves—namely, company culture.

In the mini e-book, *Changing Culture and Building Confidence*, four expert contributors offer guidance on navigating the bumpy process of rebuilding culture to embrace continuous delivery. As continuous delivery consultant Tommy Tynjä asserts in his essay, “Change is easy to introduce in an organization. The difficult part is making it stick.”

In that entry, Tynjä reminds readers that discussions around continuous delivery often center on automating tools and processes. That’s not enough, he writes. The way the entire organization sees software development must change. Everyone must understand its benefits and how the process will affect daily work. That takes more than a quick sit-down chat: it requires time, attention, and frequent reminders. In effect, Tynjä suggests, changing culture is its own kind of continuous delivery.

Mark Nelson, an architect at Oracle, notes that just because continuous delivery offers new efficiencies and capabilities doesn’t mean workers will automatically take advantage of it—or even wish to. The continuous delivery champion must commit whatever time and energy it takes to make individuals comfortable with these changes, and the process will at times be painful. That may require a hard-sell approach.

“Be steadfast on the goal,” Nelson writes, “do not compromise on quality.” If you do, he warns, people will “take the easy way out every time.”

In his essay, BEKK continuous delivery and DevOps lead Stein Inge Morisbak shares his experience heading up a small software development project with a close-knit team. His group had full responsibility for the end product’s success or failure, so it produced software incrementally, dealing with failures as early in the process as possible.

The result? Morisbak’s team created valuable software with the fewest errors of any project he has ever been involved with. Why? Because just three roles were defined for the project team: the sponsor, the users, and the “techies.” No one else. Input from architects or project managers outside the group would merely have complicated matters, he writes. Instead, his group in effect forged its own path to continuous delivery.

The more freedom and responsibility technically skilled people have, Morisbak concludes, the more creative they will be and the more confidence they will show in continuous delivery.

These are just a few of the lessons offered in *Changing Culture and Building Confidence*. The mini e-book is the fifth installment in a larger e-book, **[Lessons from 29 DevOps Experts on the Best Way to Make the Transition to Continuous Delivery](#)**. The publication, sponsored by Zend, provides best practices and advice from DevOps industry leaders. For those who want to learn more about implementing continuous delivery, this e-book covers each step: getting started in continuous development, integrating and automating the process, getting the team on board, changing the culture, and best practices for the future. **[Download the full e-book now](#)** to take advantage of these expert insights and determine whether continuous delivery is right for your business.

- Kevin Featherly

# FOREWORD

## Exploring Continuous Delivery

Innovation has changed. Gone are the days when a solitary genius holed up in a garage conceived a big idea, and then painstakingly perfected and brought it to market years later. Today, innovation is fluid, fast moving, and collaborative. Innovation is the engine for growth and value creation in the modern world, and software is the fuel.

The ability to create new, high-quality software applications and bring them to market more quickly is the “X factor” that defines industry leaders, and these leaders all have one thing in common: their IT organizations are leaving traditional approaches behind in favor of new, agile, collaborative approaches to the design, development, and delivery of applications.

At Zend, we are committed to helping companies deliver innovation more quickly. We’ve seen the dramatic results of this trend in working with Fiat, Hearst Corporation, BNP Paribas, Newell Rubbermaid, Prada, and other customers that are achieving faster and more frequent releases of more reliable software and, as a result, improving their business growth and profitability. Like other companies around the world, their success stems from the adoption of Continuous Delivery methodologies and best practices.

This e-book has been created for companies at virtually any stage of the journey toward Continuous Delivery. In the following pages, you’ll find essays from software industry leaders whose experiences, insights, and solutions can make it a lot easier to get started, progress smoothly, and finish strong.

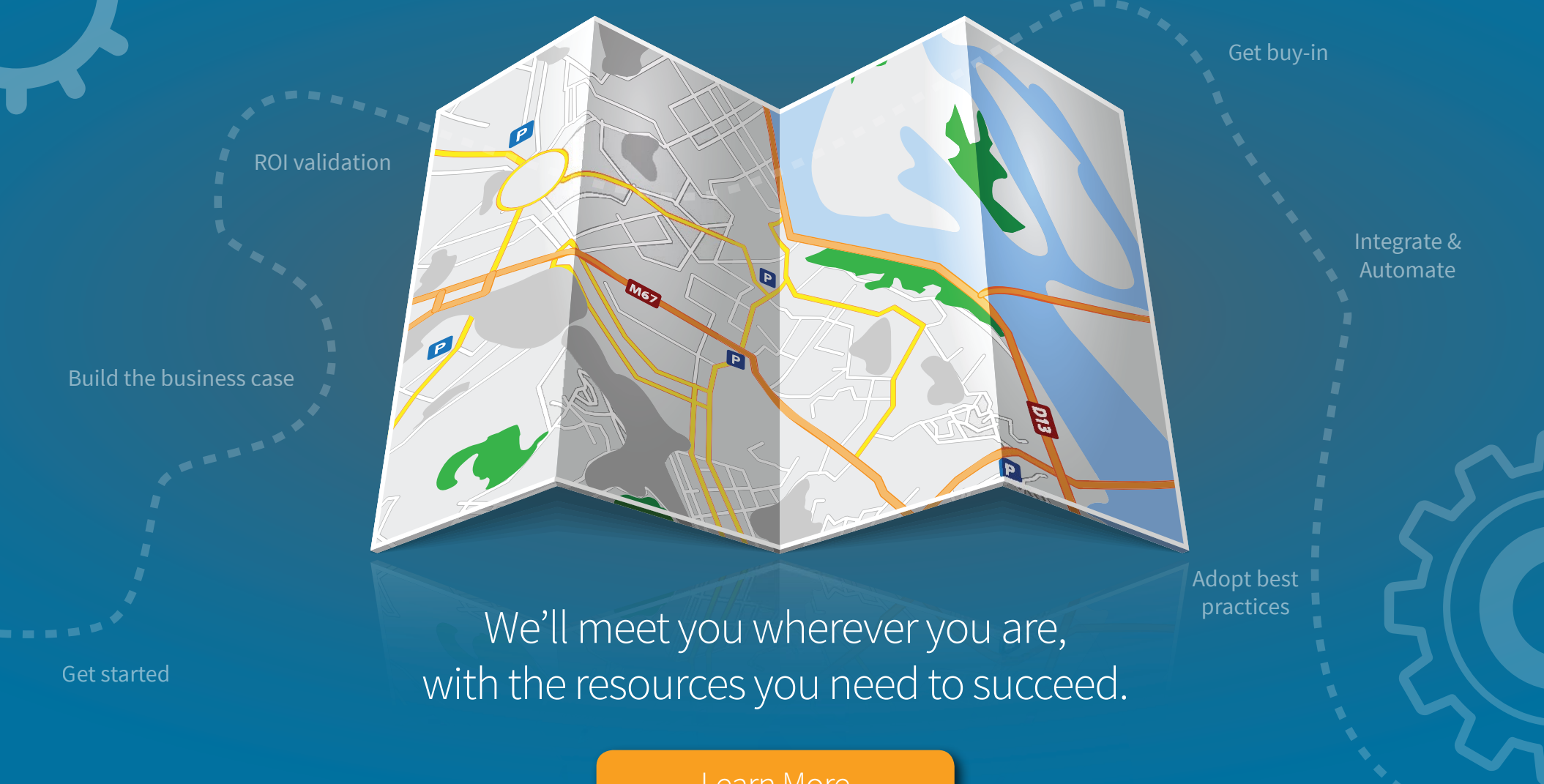


Wishing you the best success,  
Andi Gutmans  
CEO, Zend



Zend helps businesses deliver innovation more quickly, on a larger scale, and across more channels than ever before. More than 40,000 companies rely on our solutions, including Zend Server, the integrated application platform for mobile and web apps. Zend Server provides superior tools for creating high-quality code, best-in-class infrastructure for moving applications from source control through deployment, and the best back-end platform for performance at Web scale. Zend helped establish PHP, which today powers more than 240 million applications and websites around the world. Visit us at [www.zend.com](http://www.zend.com).

# Continuous Delivery is a Journey



“The ability to create new, high-quality software applications and bring them to market more quickly is the “X factor” that defines industry leaders.”

**Andi Gutmans**, CEO & Co-founder, Zend



# INTRODUCTION

Continuous Delivery isn't just a technical shift, it's a cultural one. Even though it takes hard work to make the transition, the benefits can't be ignored. Faster time to market, better quality product, competitive advantage, higher customer satisfaction and reduced cost of development are just a few of the benefits driving CD to become the new norm.

With the support of Zend, we reached out to 29 top DevOps professionals and asked them the following question:

**Your friend has been tasked with transitioning her company's software development efforts to Continuous Delivery. She's extremely capable, but she's nervous about leading the transition. Please share a story from your own experience that will provide her with a critical piece of advice that will help her to be more successful.**

The response was fantastic. Not only did we receive insightful essays, but the expert advice came from the very people who have been leading this revolution – people like Gene Kim, Andi Gutmans, Rebecca Parsons, Scott Hanselman and Andrew Yochum. The essays in this book roughly break down into six categories that range from understanding the business case for CD through actually making the journey. We hope the collective wisdom and hard-learned lessons contained in these pages will inspire you and help you take your own development efforts to a higher level.



All the best,  
David Rogelberg  
Editor

© 2014 Studio B Productions, Inc. | 62 Nassau Drive | Great Neck, NY 11021 | 516 360 2622 | [www.studiob.com](http://www.studiob.com)

# Changing Culture and Building Confidence



## **TOMMY TYNJÄ**

Getting Your Organization Excited About Continuous Development.....7



## **JAMIE INGILBY**

Small Steps Toward Full Automation.....8



## **MARK NELSON**

Avoiding the Easy Way: Selling Continuous Delivery in Your Business.....10



## **STEIN INGE MORISBAK**

The Freedom to Be Creative.....12



# GETTING YOUR ORGANIZATION EXCITED ABOUT CONTINUOUS DEVELOPMENT



## TOMMY TYNJÄ

Continuous Delivery  
Consultant at Diabol AB

Tommy Tynjä is a continuous delivery consultant and open source software contributor who is passionate about continuous delivery, test-driven development, automation, and tools that boost developer productivity and get software shipped faster. Working primarily with Java-based solutions, he has several years of developer and architect experience, ranging from small start ups to enterprises and everything in between.



Twitter



Website



Blog



**Download the full ebook: *Lessons from 29 DevOps Experts on the Best Way to Make the Transition to Continuous Delivery***

**C**ontinuous delivery is about making sure an application is always in a releasable state, where manual, repetitive, and error-prone tasks related to a release, such as setting up servers and middleware, have been fully automated. Many discussions on continuous delivery focus on the automation and tooling aspects, but it's important to remember that continuous delivery requires more than just tools and products to succeed. To successfully implement continuous delivery, you need to change the culture of how an entire organization views software development efforts.

Change is easy to introduce in an organization. The difficult part is making it stick. This is especially true when starting an effort to introduce continuous delivery in an organization, whether it's imposed by management or an initiative from developers tired of their unproductive, old-fashioned way of working. Every member the initiative affects must understand the benefits of the implementation and how it will affect the way they do their daily work. This is typically not done through one or a couple of formal meetings, where the organization's management presents the continuous delivery efforts. This is something that has to be done continuously; it has to be rooted in every employee working in the software development departments.

It is also important that all employees feel that they have the proper forum in which to discuss how continuous delivery will affect their daily ways of working. Even after continuous delivery has been introduced, team members will need reminders of why, for example, manual changes to environments are bad or how shortcutting the deployment pipeline negatively affects the overall quality of the system and undermines the continuous delivery principles as a whole.

The biggest boost an organization can get when implementing continuous delivery is to get people excited and passionate about it.

***“To successfully implement continuous delivery, you need to change the culture of how an entire organization views software development efforts.”***

## KEY LESSONS

- 1 CONTINUOUS DELIVERY REQUIRES MORE THAN JUST TOOLS AND PRODUCTS TO SUCCEED: YOU MUST CHANGE THE ORGANIZATION'S CULTURE.**
- 2 EVERY MEMBER THE INITIATIVE AFFECTS MUST UNDERSTAND THE BENEFITS OF THE IMPLEMENTATION AND HOW IT WILL AFFECT THE WAY THEY DO THEIR DAILY WORK.**

# SMALL STEPS TOWARD FULL AUTOMATION



## JAMIE INGILBY

Software Development  
Manager

Jamie Ingilby is a software development manager based in the United Kingdom. He has worked in many large and small technology organizations, supporting Web services. He is passionate about software development and is a huge advocate of continuous delivery as a means of producing quality software at pace with confidence. He enjoys good coffee and tea, particularly Earl Grey.



Twitter



Website



Blog



**Download the full ebook: *Lessons from 29 DevOps Experts on the Best Way to Make the Transition to Continuous Delivery***

worked in a large organization that operated in a heavily regulated environment where uptime was critical. Conventional wisdom had embedded a mindset that fewer changes meant lower risk, while the business wanted to become more reactive to the marketplace.

A small number of pilot delivery teams were tasked with moving toward a continuous delivery approach, but the teams initially found it difficult to gain traction. Automated functional testing and deployment were easy to get started with, but the teams found it difficult to automate for such parts of the pipeline as security, configuration, and infrastructure, which other internal teams historically owned.

Continuous delivery requires cultural adoption of the principles across the organization. This is especially important in areas of the business not traditionally used to adopting the continuous delivery mindset. Specialist infrastructure teams, for example, were still used to putting major changes expected to boost performance in production first. Our initial approach was to try to obtain buy-in from across the organization by reinforcing the benefits and principles of continuous delivery, but

it's easy for people to brush this message off without real cultural change. We realized that empowerment and ownership with those who encounter the pain of manual processes would help. We built role-combined delivery teams that were encouraged to take full ownership of their own delivery pipelines and that worked to automate as much as possible. Small steps toward continuous delivery in fully automated testing and deployment, helped to increase the frequency of and our confidence in our releases.

As we refined our immature pipeline, it helped to shine a spotlight on areas that were outside the scope of the existing pipeline; teams began to challenge the existing processes. The quick wins and early steps had helped to initiate a cultural shift, which in turn helped all areas of the organization understand what we were trying to achieve. When we had a common understanding of what we wanted the result to be, we were able to consider new approaches for automating the parts that remained. Continuous delivery in some environments can be a long road; don't be afraid to start with small, incremental steps that demonstrate value, and use that to build momentum.

*“Small steps toward continuous delivery in fully automated testing and deployment, even with manual steps built into the process, helped to increase the frequency of and our confidence in our releases.”*

## KEY LESSONS

- 1 CONTINUOUS DELIVERY REQUIRES CULTURAL ADOPTION OF THE PRINCIPLES ACROSS THE ORGANIZATION.**
- 2 EMPOWERMENT AND OWNERSHIP WITH THOSE WHO ENCOUNTER THE PAIN OF MANUAL PROCESSES HELPS.**
- 3 AS YOU REFINE YOUR PIPELINE, SHINE A SPOTLIGHT ON AREAS THAT ARE OUTSIDE THE SCOPE OF THE EXISTING PIPELINE.**



# Bring your code and user feedback closer together



Intuit founder **Scott Cook** is an advocate for a “rampant innovation culture” and allowing employees to do rapid, high-velocity experiments. Several years ago Intuit’s Consumer Division took this to heart, and transformed the TurboTax website through **Continuous Delivery**.

## The result?

They ran 165 experiments during the 3-month tax season. The website saw a **50% increase** in the conversion rate. The employees **loved it** because they saw their **ideas come to market**.

---

**Gene Kim**, Author and Researcher, IT Revolution Press discusses success through DevOps practices.



# AVOIDING THE EASY WAY: SELLING CONTINUOUS DELIVERY IN YOUR BUSINESS



**MARK NELSON**

Architect at Oracle

Mark Nelson is an architect on the Platform Architecture team in Oracle Development. Mark's focus is continuous delivery, configuration management, and provisioning, making it simple to manage the configuration of complex environments and applications built with Oracle Database, Fusion Middleware, and Fusion Applications on-premises and in the cloud. Before joining the Platform Architecture team, Mark had been a senior member of the A-Team since 2010 and has worked in the IT industry since 1994.



**Download the full ebook: [Lessons from 29 DevOps Experts on the Best Way to Make the Transition to Continuous Delivery](#)**

**W**hen you introduce continuous delivery into your company, you're giving people a new capability that will save them time and improve reliability, visibility, repeatability, and quality. But that does not mean that they will use it—or even want to.

No matter how significant the benefits, no matter how well you demonstrate and quantify those benefits, people tend to be resistant to change. You should be prepared to “sell” it to all of the stakeholders and to take an active, hands-on role—to sit with people, one after another, and help each one develop the understanding and comfort he or she needs to make the investment of time to move to something new.

You're also going to shine a light on and possibly exacerbate many existing problems in your current build-test-release processes—even in your source and binary management processes. It's easy for people to blame these “problems” on the new approach, certainly much easier than admitting that the issues were always there. Introducing continuous delivery provides an opportunity to go back and correct some of those problems from the past. Many of today's issues are steps that were taken for perfectly valid tactical reasons, but somehow no one ever got around to replacing them with something more strategic.

Be steadfast on the goal: do not compromise on quality. If you relax the rules, even a little, and let people find a way around things that are difficult, they will take the easy way out every time. Remember, one of the key tenets of continuous delivery is to bring the things that are difficult forward in the project to reduce risk.

It's going to be a bumpy ride, so keep a firm hand on the wheel.

**“Introducing continuous delivery provides an opportunity to go back and correct some of those problems from the past.”**

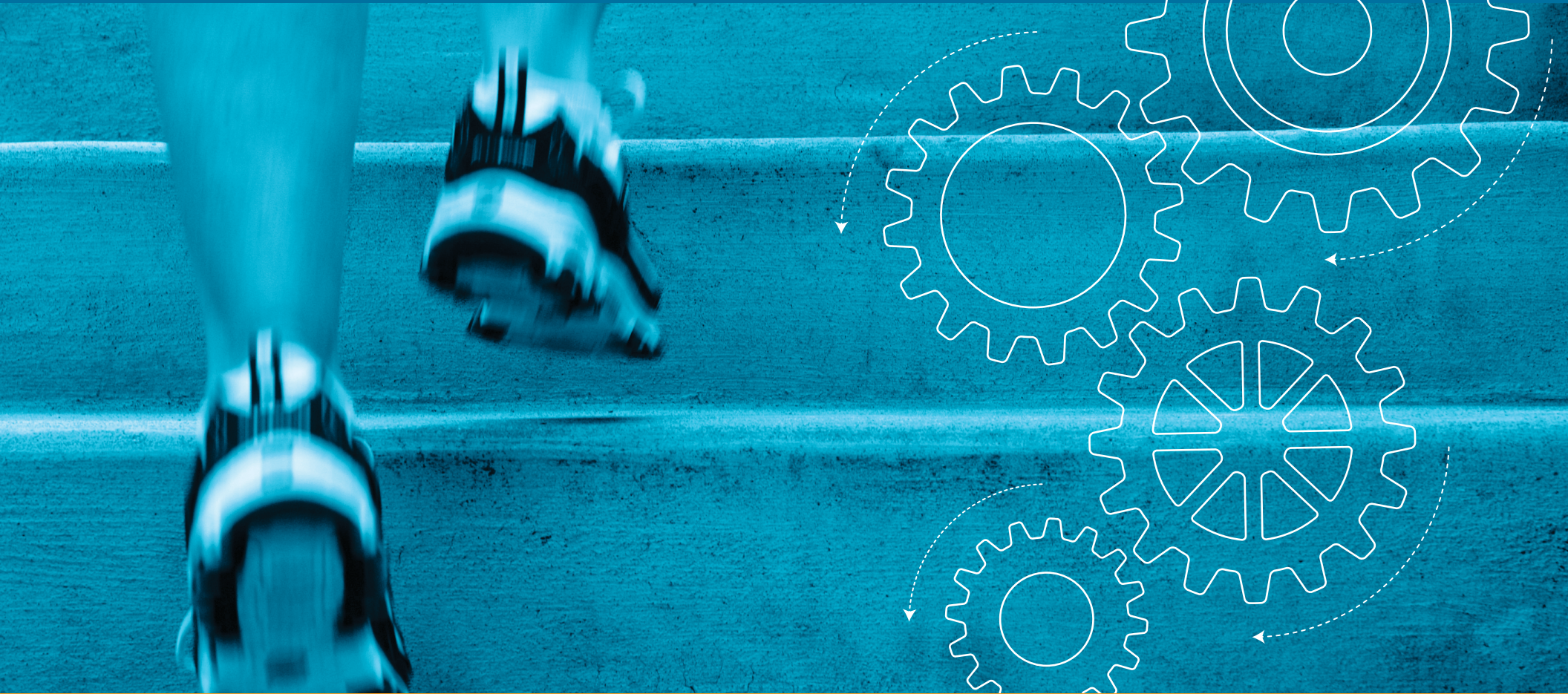
## KEY LESSONS

- 1 BE PREPARED TO “SELL” CONTINUOUS DELIVERY TO STAKEHOLDERS.**
- 2 DON'T COMPROMISE ON QUALITY OR RELAX THE RULES.**



# Continuous Delivery

Six Steps to Faster Releases without Breaking Anything



More Innovation • Better Quality • Earlier Feedback • Faster Releases

Start off on the right foot.

Read the White Paper ►



# THE FREEDOM TO BE CREATIVE



## STEIN INGE MORISBAK

Manager at BEKK Consulting AS

Stein Inge Morisbak is practice lead for continuous delivery and DevOps at BEKK. He is a true agile evangelist, with 15 years of experience contributing to and helping others to become better at producing excellent software together with demanding customers. He is also an experienced speaker at conferences and the founder of DevOps Norway Meet-Up.



Twitter



Website



Blog



**Download the full ebook: [Lessons from 29 DevOps Experts on the Best Way to Make the Transition to Continuous Delivery](#)**

Ever since I started in software development, I have worked with skilled and highly educated colleagues. They have been product owners, project managers, architects, testers, systems administrators, and programmers. Having all these roles on a project, you would think that every aspect of a software project would be covered and that we would succeed every time. Yet our projects failed, as unfortunately most software projects do.

At some point, I was asked to be tech lead on a smaller project. The project sponsor wanted to follow the project closely, so we formed a close-knit team around her. I chose seven other people for the project and asked for servers to host the software we were going to build. The project was amazingly successful. We delivered a valuable product and with fewer errors than anything I had been involved with before. Adding new features was easy, and the project sponsor was extremely satisfied when she saw her ideas running in production soon after they were born. We were practicing continuous delivery.

The reason we were able to deliver high-quality software with great value continuously was not that we were superhuman developers. I realized that the main reason for our success was that we only had three roles: the sponsor, the users, and the “techies.” I found out that any roles involved in a project that do not directly contribute toward the goal of putting valuable software in the hands of users as quickly as possible should be carefully considered. We were not being continually measured by project managers or pushed beyond our limits. Architects were not telling us what kind of products or patterns to use. Testers did not obstruct our changes from getting into production. Systems administrators were not trying to prevent changes that could destabilize their servers. We had to take responsibility for the testing ourselves, and we worried about the infrastructure because we were the only ones who could be blamed. We delivered software incrementally to fail as early as possible so that we could correct errors before they got out of hand and get a feel for what the users really wanted.

Technically skilled people are used to solving challenges. The more freedom and responsibility they have, the more creative they will be—as long as their goal is known: to satisfy the sponsors through early and continuous delivery of software valuable to users.

*Any roles involved in a project that do not directly contribute toward the goal of putting valuable software in the hands of users as quickly as possible should be carefully considered.*

## KEY LESSONS

- 1 CAREFULLY CONSIDER ANY ROLES INVOLVED IN A PROJECT THAT DO NOT DIRECTLY CONTRIBUTE TO THE GOAL.**
- 2 DELIVER SOFTWARE INCREMENTALLY TO FAIL AS EARLY AS POSSIBLE SO THAT YOU CAN CORRECT ERRORS BEFORE DEPLOYMENT.**

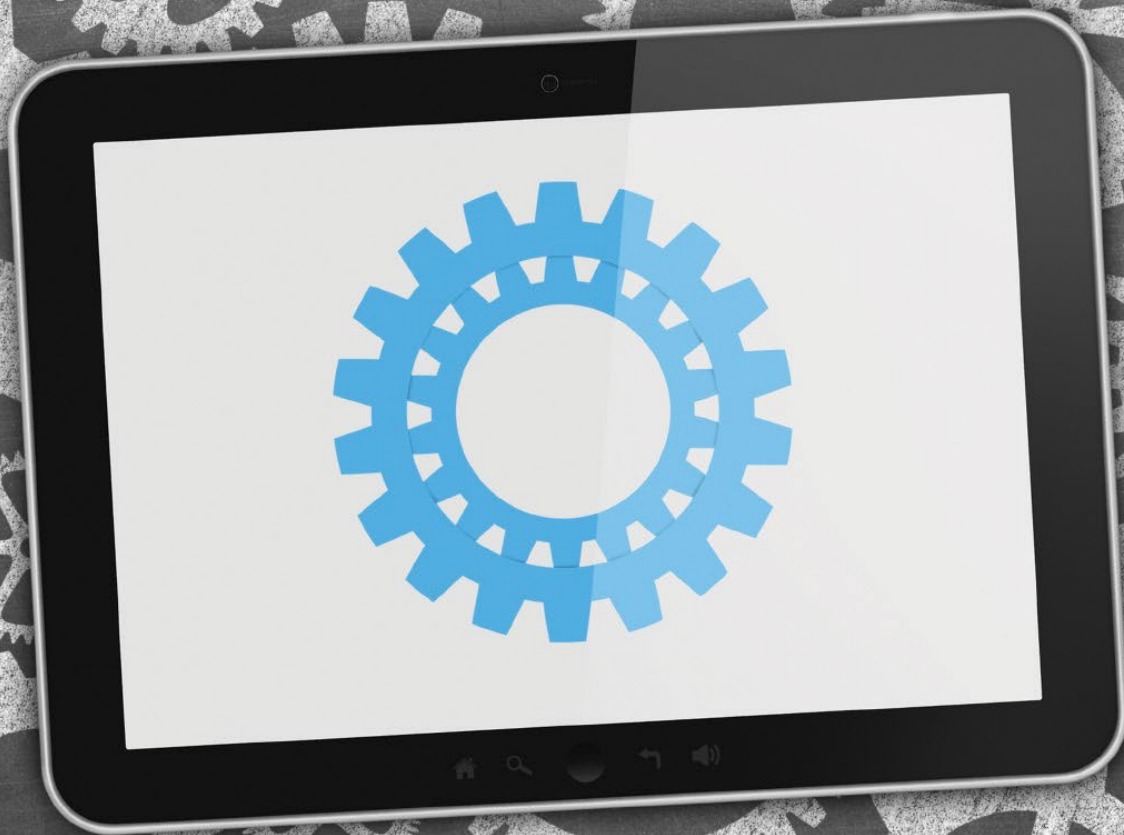


From the front lines  
of Continuous Delivery  
direct to your screen

Take 10  
minutes

Learn from 29  
DevOps  
experts

Get a 360°  
view of the  
journey  
to Continuous Delivery



**Continuous Delivery:** Lessons from 29 DevOps Experts on  
the Best Way to Make the Transition to Continuous Delivery

**Download eBook** ►

