

Getting Started

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



Sponsored by:



GETTING STARTED

Continuous delivery is transformative, with every component of software delivery automated. The silos among development, quality assurance (QA), and operations teams are dismantled, and communication increases exponentially. It seems a no-brainer. Nonetheless, it's a scary proposition for a lot of companies—not least among their concerns is how to get started.

That question is addressed in *Getting Started*, a mini-e-book written by five DevOps experts and advocates. It helps those interested in continuous delivery and DevOps to wrap their minds around taking those crucial first few steps. *Getting Started* is the second in a series of six mini-e-books sponsored by Zend, each offering unique insights into continuous delivery.

Maurice Kherlakian, lead architect for continuous delivery at Zend, emphasizes getting relevant stakeholders on board. “The continuous delivery paradigm is a big departure from a waterfall methodology,” he writes. “New tools are needed.” Enable collaboration across all functional boundaries, he suggests, and automate wherever possible.

Another key theme is starting small and expanding process improvements incrementally. Alex Schwartz, a DevOps evangelist at HERE, a Nokia business, calls it “aggressively using the baby steps technique.” He cites an example from his own experience in which QA, release management, and operations functions were merged. The process started with a small, crucial task—determining the best ratio of overhead to potential time savings, which was implemented successfully over several days. Other small improvements followed, each given high priority. Performance naturally improved. “Along the way,” he writes, “we established our belief in improvements, so we built trust in our ability to turn the situation around.”

Olivier Jacques, a distinguished technologist at HP and—tellingly—a working musician, makes a compelling case for implementing a deployment pipeline in which DevOps components are broken down into stages. He draws inspiration from the Depeche Mode song, “Pipeline,” which he quotes several times in his essay. “The deployment pipeline is your backbone—your assembly and delivery line,” Jacques writes. “Code changes, infrastructure changes, and monitoring changes all flow through the deployment pipelines.”

Continuous delivery requires a radical culture shift, one that won't happen overnight. Mathias Meyer, CEO at Travis CI, urges readers to get feedback from their teams to find out what holds the company back from faster releases. “Start by asking them why and how you can improve the situation to give them more confidence,” Meyer writes. “You'll find that beyond encouraging people, feature flags and thorough monitoring help a lot to get code to production quickly.”

Getting Started is one chapter of a larger e-book, [Lessons from 29 DevOps Experts on the Best Way to Make the Transition to Continuous Delivery](#). The publication provides best practices and advice from the top DevOps industry leaders. If you're interested in learning more about how to implement continuous delivery, this 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.



Continuous Delivery is a Journey



Get buy-in

Integrate &
Automate

Adopt best
practices

ROI validation

Build the business case

Get started

We'll meet you wherever you are,
with the resources you need to succeed.

[Learn More](#)

“ 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

Getting Started



MAURICE KHERLAKIAN

In the Driver's Seat with Continuous Delivery.....7



OLIVIER JACQUES

Laying On a Pipeline: Implementing DevOps.....9



FREDERIC RIVAIN

Managing the Change.....12



ALEX SCHWARTZ

Baby Steps to Improvement.....13



MATHIAS MEYER

Asking Questions to Move Toward Continuous Delivery.....15



IN THE DRIVER'S SEAT WITH CONTINUOUS DELIVERY



MAURICE KHERLAKIAN

Lead Architect for Continuous Delivery at Zend

Maurice Kherlakian is Zend's lead architect for Continuous Delivery and a seasoned PHP professional who has more than a decade of experience building and maintaining large-scale applications. He has also worked as a systems administrator for Windows and Linux platforms. As a key member of the Zend Professional Services team, Maurice helps customers optimize their use of PHP and Zend solutions. He specializes in PHP and Zend Framework architectural and performance audits and implementation of best practices.



Twitter



Website



Blog



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

Working with developers, I've seen firsthand the transformation that Continuous Delivery brings to their projects. Being able to iterate process and product changes with true agility puts them in the driver's seat, with more time for innovating and less time mired in backlogs and troubleshooting.

Even so, IT organizations that have traditionally been bound by processes, procedures, and workflows tend to see Continuous Delivery as a wholesale process change that will destabilize the IT environment and wreak havoc on their goals. They assume that a big-bang approach is required, but a phased approach to Continuous Delivery is not only preferable, it's infinitely more manageable.

Six Steps to Continuous Delivery

- 1. Start Small.** This holds true whether your organization is embarking on its first Continuous Delivery effort or updating a legacy practice. Choose a project or work group in which change is manageable, goals can be set, and results can be measured.
- 2. Get Buy-In.** Make sure the development and operations teams are on board. The Continuous Delivery paradigm is a big departure from a waterfall methodology: new tools are needed.
- 3. Implement Continuous Integration.** Implement a continuous integration solution along with automating application deployment to testing and staging environments and implementing a basic set of automated unit tests that occur with every build.

“A phased approach to continuous delivery is not only preferable, it's infinitely more manageable.”

KEY LESSONS

- 1 WHEN STARTING THE CHANGE TO CONTINUOUS DELIVERY, START SMALL AND EXPAND SLOWLY.**
- 2 GET ALL THE RELEVANT STAKEHOLDERS ON BOARD EARLY.**
- 3 AUTOMATE WHEREVER POSSIBLE.**



IN THE DRIVER'S SEAT WITH CONTINUOUS DELIVERY



MAURICE KHERLAKIAN

Lead Architect for Continuous
Delivery at Zend

Maurice Kherlakian is Zend's lead architect for Continuous Delivery and a seasoned PHP professional who has more than a decade of experience building and maintaining large-scale applications. He has also worked as a systems administrator for Windows and Linux platforms. As a key member of the Zend Professional Services team, Maurice helps customers optimize their use of PHP and Zend solutions. He specializes in PHP and Zend Framework architectural and performance audits and implementation of best practices.



Twitter



Website



Blog



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

- 4. Reinforce Best Practices, and Automate the Deploy-to-Production Pipeline.** A unit test must accompany every piece of code to ensure code functionality. Ideally, these unit tests will be automatically executed with every build, and builds will happen frequently. The entire team, including business management, should meet regularly to review which processes are working well and what needs improvement.
- 5. Adopt Automation Across the Delivery Process.** Employ zero-touch continuous builds, automated system deployments, and comprehensive automated testing. Metrics, performance monitoring, and performance analytics are available to the entire team.
- 6. Build on Your Successes.** Even if you start small, you can begin to reap the benefits of Continuous Delivery before a fully automated process is implemented and realize increasing benefits as your process matures. Compare your progress against release and quality metrics to see the incremental improvements along the way.

Continuous Delivery is an engine for innovation. In my experience, when developers get behind the wheel and learn the rules of the road, they can do amazing things.

KEY LESSONS

- 1 WHEN STARTING THE CHANGE TO CONTINUOUS DELIVERY, START SMALL AND EXPAND SLOWLY.**
- 2 GET ALL THE RELEVANT STAKEHOLDERS ON BOARD EARLY.**
- 3 AUTOMATE WHEREVER POSSIBLE.**



[Blog](#)



[Webinars](#)



[Newsletter](#)



[Chat with a Zender](#)

LAYING ON A PIPELINE: IMPLEMENTING DEVOPS



OLIVIER JACQUES

HP IT-Distinguished
Technologist at Hewlett-Packard

Olivier Jacques has more than 17 years of experience in software development, testing, and quality. His focus is on designing end-to-end solutions for application development that span multiple tools and processes, helping to keep HP's engineering in the "flow." He is a strong advocate of modern methods of developing and releasing applications through agile and DevOps strategies, applying them to massive-scale enterprises.



Twitter | Website



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

One of my passions is music. I love all kind of music, really, but I especially love electronic music. I have a band, too: we write music and lyrics and even tour Europe. Regardless of whether you like electronic music, I'm sure you have heard about one of the most successful electro-pop bands ever: Depeche Mode. In 1983, they released an album called "Construction Time Again," which features a track called "Pipeline." There's a lot to like about that track, including the massive use of sampled sounds—quite revolutionary back then. It's 2014, and you want to know everything about pipelines for your move to DevOps.

"We're laying on a pipeline."

In continuous delivery and DevOps, an important concept is the *deployment pipeline*. The deployment pipeline is your backbone—your assembly and delivery line. Code changes, infrastructure changes, and monitoring changes all flow through the deployment pipelines. You can allow some changes, deny others, or reroute them. You, supported by intelligent tests and automation, are always in control.

- **Tools.** "Get out the crane/Construction time again." Recent technology evolutions, and specifically cloud computing, have made deployment pipelines even more relevant in an IT world. We are now able to treat infrastructure (compute, storage, network) as code and benefit from the same processes that have been applied to source code for years, such as team collaboration and change management (including traceability, peer reviews, defect management, and automated testing). The deployment pipeline harnesses all the tools in a meaningful and visible way.

"To implement DevOps, you need to enable collaboration across functional boundaries. Here, people are key."

KEY LESSONS

- 1 THE DEPLOYMENT PIPELINE HARNESSSES ALL THE TOOLS IN A MEANINGFUL AND VISIBLE WAY.
- 2 AUTOMATE MOST BUT NOT NECESSARILY ALL OF THE PROCESS.
- 3 ENABLE COLLABORATION ACROSS FUNCTIONAL BOUNDARIES.



Download the
eBook



LAYING ON A PIPELINE: IMPLEMENTING DEVOPS



OLIVIER JACQUES

HP IT–Distinguished
Technologist at Hewlett-Packard

Olivier Jacques has more than 17 years of experience in software development, testing, and quality. His focus is on designing end-to-end solutions for application development that span multiple tools and processes, helping to keep HP's engineering in the "flow." He is a strong advocate of modern methods of developing and releasing applications through agile and DevOps strategies, applying them to massive-scale enterprises.



Twitter | Website



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

- **Process.** “On this golden day/work’s been sent our way.” A deployment pipeline leads to a process that is not only faster but safer. You’re looking to automate most but not necessarily all of the process. Deployment procedures, controls, tests, triggers, reactions to monitoring alerts—the deployment pipeline structures your process, helps you communicate it, and (most importantly) improve it.
- **People.** “From the heart of the land to the mouth of the man.” To implement DevOps, you need to enable collaboration across functional boundaries. Here, people are key. Deployment pipelines enable collaboration, continuous improvement, and lower the walls between the boundaries. Everything becomes visible: the changes going through the pipeline, the triggers, the feedbacks, the gates. Deployment pipelines bring people together.

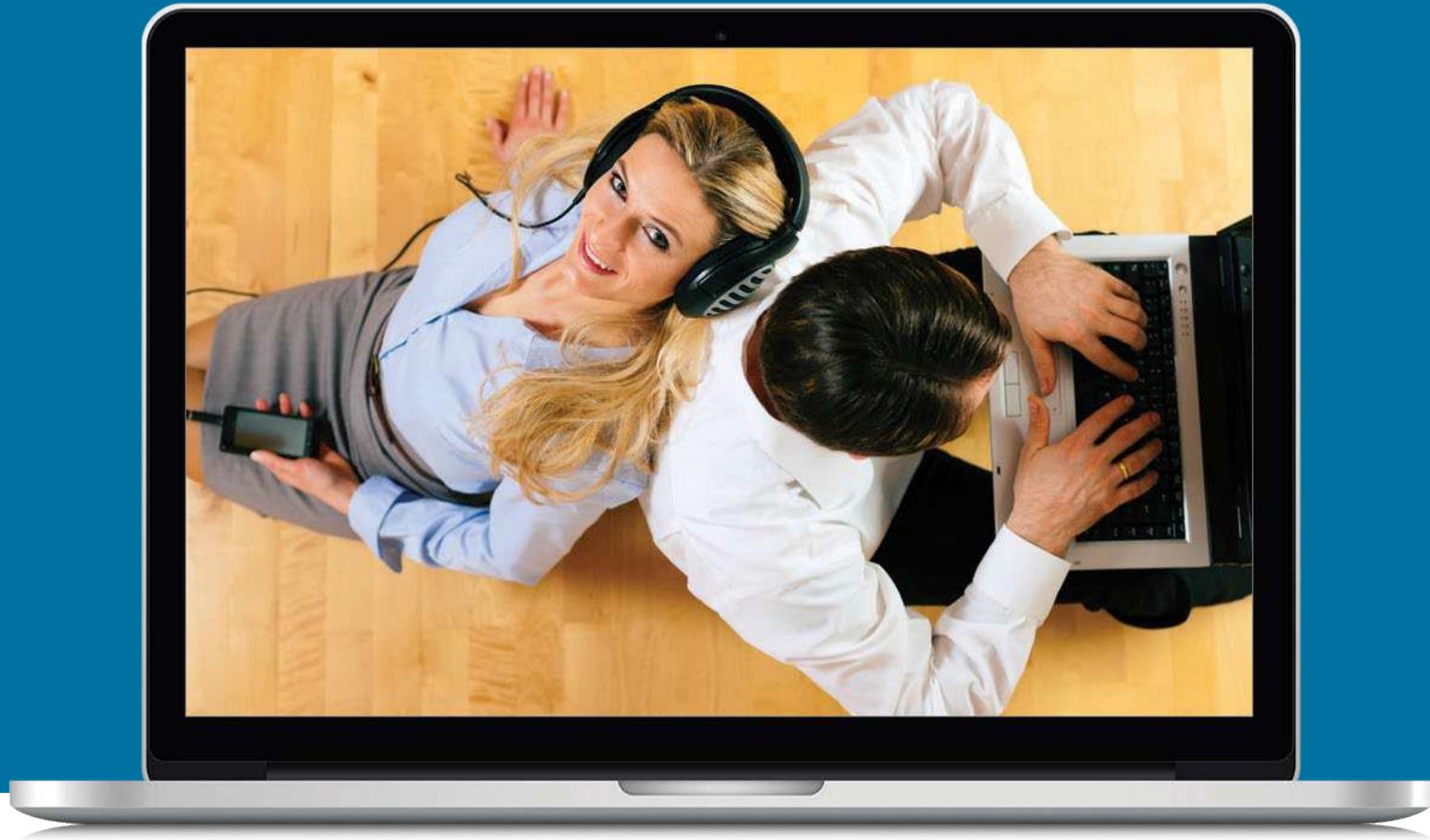
You want to implement DevOps? Then, you want to know everything about deployment pipelines.

KEY LESSONS

- 1 **THE DEPLOYMENT PIPELINE HARNESSSES ALL THE TOOLS IN A MEANINGFUL AND VISIBLE WAY.**
- 2 **AUTOMATE MOST BUT NOT NECESSARILY ALL OF THE PROCESS.**
- 3 **ENABLE COLLABORATION ACROSS FUNCTIONAL BOUNDARIES.**



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**.

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

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**.



MANAGING THE CHANGE



FREDERIC RIVAIN

Head of Software
Development at
Betclic Everest Group

Frederic Rivain is a dedicated, highly motivated chief technical officer and head of software development at Betclic. He has extensive expertise in content and media—mobile, television, or Web—from e-commerce to gaming.



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

I have traveled the path to DevOps and continuous delivery with various levels of success. I have come to realize that this is not only about tools, scripts, and automation (which is in reality the easy part) but above all about a certain state of mind and change management.

As strange as it may seem, the benefits of continuous delivery are not obvious for many people. Most prefer to rely on what they're used to—even tedious, repetitive tasks—because this is the way it has always been. So, you will first need to gain the trust of your team and get them to agree to the vision.

This can be tricky, depending on the actual organization and the size of the company. You may want to start with a few pioneers on an isolated project, where you will be in a position to run a pilot to demonstrate how the new system can work.

Showing a strong success and visible benefits is key to getting others to agree to try your way of doing things. I also encourage you to rely on your business team as sponsors, because they will be your fiercest allies and the first to advocate all the advantages of continuous delivery (after they start reaping the rewards).

Then, you can start to expand across the whole organization. Use your pilot team to explain and train other willing people, and spread the practice until you have reached a critical point where the rest of the company more naturally joins the trend.

Of course, you will have ups and downs: this is a journey with turns and bumps, but once again, the secret is that this is all about a state of mind and managing the change in the company.

“Showing a strong success and visible benefits is key to getting others to agree to try your way of doing things.”

KEY LESSONS

- 1 FIRST, GAIN THE TRUST OF YOUR TEAM, AND GET THEM TO AGREE TO THE VISION.**
- 2 START WITH A FEW PIONEERS ON AN ISOLATED PROJECT THAT YOU CAN USE AS A PILOT.**
- 3 RELY ON YOUR BUSINESS SPONSORS.**

BABY STEPS TO IMPROVEMENT



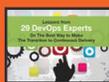
ALEX SCHWARTZ

DevOps Evangelist at HERE,
a Nokia business

Alex Schwartz is a speaker, coach, and lean manager who focuses on DevOps, continuous delivery, and agile testing. His journey in the industry started 20 years ago as a programmer; over the years, he assumed different roles in different organizations. Most recently, he has been working in the trenches for eBay and later for Nokia to introduce DevOps and continuous delivery to those organizations.



Twitter | Website



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

Imagine yourself stuck in firefighting mode, with no time left for necessary improvements. In such cases, the “baby steps” technique can be your good friend.

Here’s an example. A few years ago, I joined Nokia to handle release management for a dozen Representational State Transfer (REST)-ful services. I learned quickly that releasing was non-trivial, integration testing was complex and time consuming, and deployment automation was rudimentary. As a first step, my team decided to reduce handovers by merging the quality assurance, release management, and operations teams for those services. The resulting delivery team owned the value stream to production.

During the first month, we spent more than 80 percent of our time resolving problems. Within 12 months, we spent 10 percent of our time on unplanned work. Meanwhile, we increased the number of releases by 20.

We achieved this progress mainly by aggressively using the baby steps technique. Because we didn’t have time to come up with a huge roadmap of improvements, we never created one. A big roadmap would actually have caused more harm than good, inducing more stress. After we extinguished one of our usual fires, we took a break, had a coffee, and afterwards as a little celebration we spent 10 minutes coming up with three small improvement ideas. After selecting the one with best potential—that is, the best ratio of overhead versus potential time savings—we implemented it over the next few days. We assigned a high priority to this task; because it was tiny (two hours maximum), we actually managed to resolve it before the next fire broke out.

To ensure that the baby step was tiny, we relaxed some of the usual rules for development. For example, we allowed a bit more code duplication and removed that duplication later. In favor of the KISS principle (i.e., “keep it simple, stupid”), we also tried hard to avoid generic, configurable code and used hard-coding as much as possible. Step by step, we improved our performance, our automation, and our product. Along the way, we established our belief in improvements, so we built trust in our ability to turn the situation around.

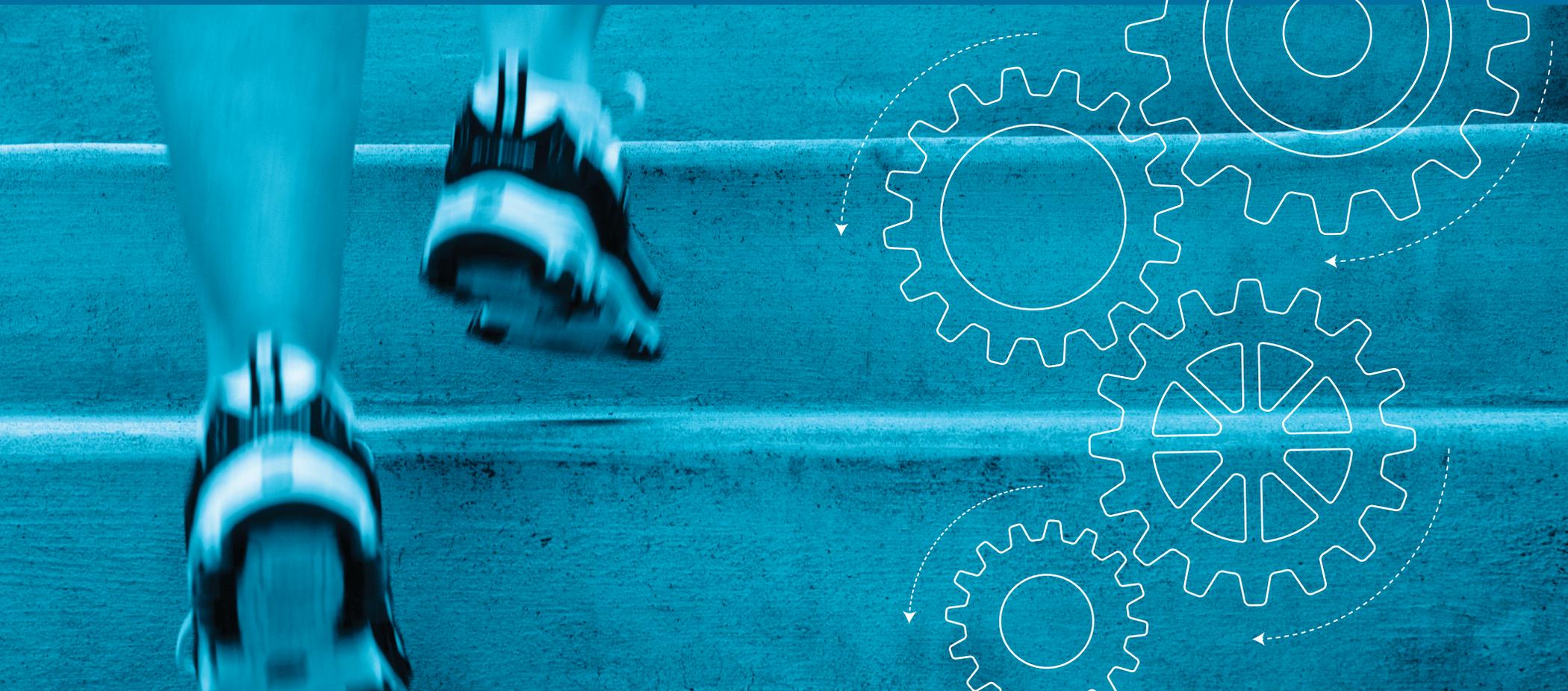
“ To ensure that the baby step was tiny, we relaxed some of the usual rules for development. For example, we allowed a bit more code duplication and removed that duplication later. ”

KEY LESSONS

- 1 **USE “BABY STEPS” TO IMPLEMENT SMALL BUT CRUCIAL CHANGES.**
- 2 **GIVE YOUR IMPROVEMENTS HIGH PRIORITY.**

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 ►



ASKING QUESTIONS TO MOVE TOWARD CONTINUOUS DELIVERY



MATHIAS MEYER
CEO of Travis CI

Mathias Meyer is the CEO and “Happiness Officer” at Travis CI. He has an interest in infrastructure, distributed systems, bacon, and coffee—the latter in unnatural amounts.



Twitter



Website



Blog



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

Continuous delivery is a culture shift in most teams. You go from big, weekly, biweekly, even monthly releases to shipping in small increments. Rather than focus on big tasks, teams break them down into smaller pieces, shipping them whenever they’re ready directly to production.

This flow poses a challenge for teams coming from the “old way” of shipping. Just like testing and continuous integration, such a transition won’t happen overnight, but gentle nudges in the right direction will help foster it.

Start by focusing on how you can find out what’s keeping your team from shipping sooner and faster. How can you make the build fast enough to get quick feedback? How can you improve your deployment process to get changes to production quickly? How can you ship new changes without breaking existing features for your customers?

Continuous delivery starts with asking “how?,” with asking your team questions. People tend to be worried about working on small changes, shipping them quickly. Start by asking them why and how you can improve the situation to give them more confidence. You’ll find that beyond encouraging people, feature flags and thorough monitoring help a lot to get code to production quickly. Together with a thorough and fast test suite, they give you confidence.

Confidence is the key to implementing continuous delivery successfully. There will always be doubts, but working together, you can find solutions to remove them. With enough confidence, your team can focus on what’s important—customer happiness, fixing bugs, adding new features gradually but confidently.

When you sit down with your team and start to ask questions, bring up ideas, and ask for feedback, you can start moving toward continuous delivery.

“You’ll find that beyond encouraging people, feature flags and thorough monitoring help a lot to get code to production quickly.”

KEY LESSONS

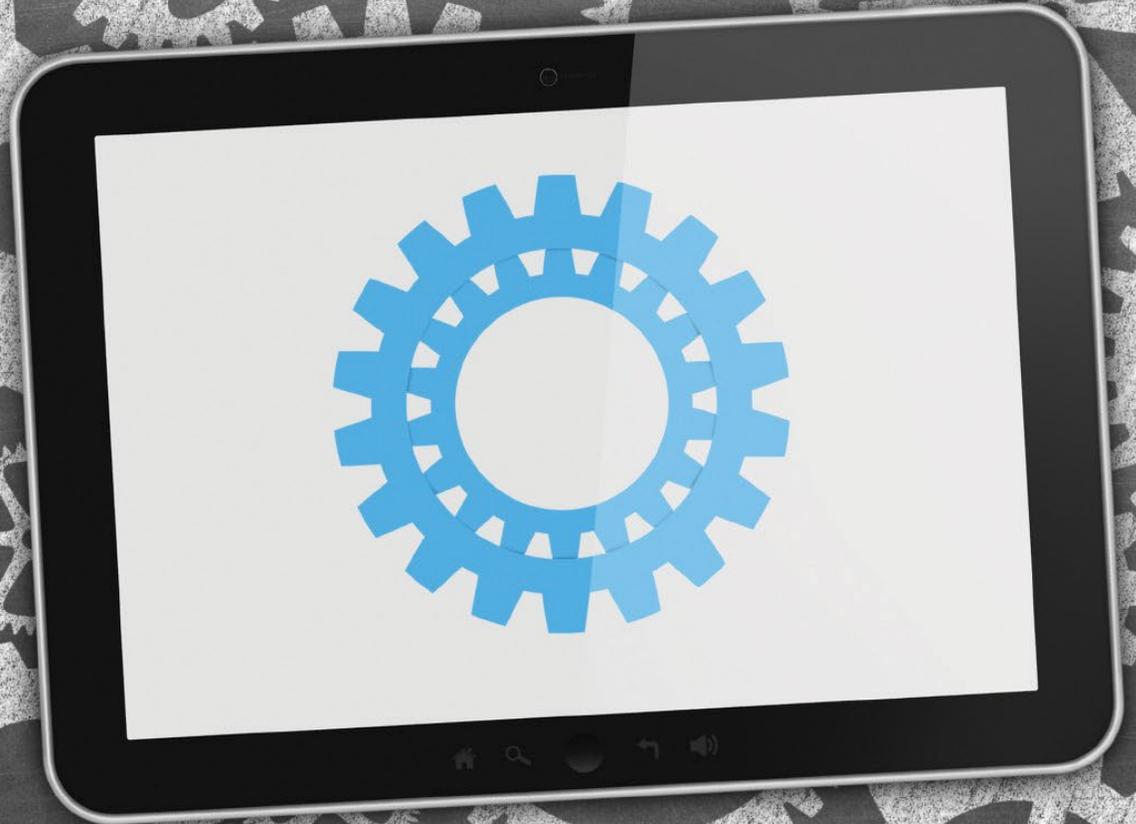
- 1 ASK YOUR TEAM QUESTIONS, ESPECIALLY WHY THEY’RE CONCERNED WITH WORKING ON SMALL CHANGES.**
- 2 DO WHAT YOU CAN TO BUILD CONFIDENCE IN YOUR TEAMS.**
- 3 SEEK FEEDBACK FROM YOUR TEAMS.**

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](#) ▶

