What is DevOps?

CloudBees.

Contents

Introduction

It seems that everyone in IT these days is talking about DevOps. From conferences to articles and books, the term DevOps has taken the IT world by storm. The buzz is understandable, as many IT groups are looking for a way out of the morass of delayed projects, questionable quality and missed deliveries in which they often find themselves.

It's clear that DevOps has the potential to address the many challenges that IT faces. Organizations that have embraced DevOps – including companies such as Etsy, Netflix, Target, Walmart, Amazon and Facebook[1] have shown that DevOps principles can lead to competitive differentiation by enabling teams to deliver higher quality software at the speed of ideas. As a result, it's not a stretch to say that DevOps is becoming the industry standard for software development.

With all this excitement, though, comes a sneaking suspicion that not everyone is talking about the same thing when they talk about DevOps. This suspicion is reinforced by CTOs who claim they are "doing" DevOps or vendors selling DevOps tools that magically enable you to "do" DevOps. It can be helpful to reconcile the many interpretations of DevOps that muddy the water and potentially inhibit adoption.

Since the earliest days of DevOps, debates about what DevOps is have existed. Fortunately, over the past few years there appears to be a growing consensus. In this whitepaper, we take a look at:

- What DevOps is
- Why you must care about it (hint: your business depends on it), and
- Whether it's a passing fad or here to stay

So What is DevOps, Really?

To clearly understand what DevOps is, it helps to understand what DevOps is not. DevOps is not a methodology or process, nor is it a single tool or technology. In fact, DevOps cannot even be defined as being strictly about development and operations. And, although many of the organizations well-known for their success with DevOps are Software as a Service (SaaS) companies, DevOps is certainly not only for SaaS applications. Lastly, DevOps is most definitely not something you "do."

Recent consensus on what DevOps is centers on the idea that DevOps is primarily about culture. The DevOps culture is based on a set of principles an organization initially aspires and ultimately adheres to. Organizations that have adopted this culture value collaboration, experimentation and learning. In a DevOps culture all participants in the software delivery lifecycle (not just development and operations) align around a shared goal: the rapid delivery of stable, high-quality software from concept to customer. Since DevOps is a cultural thing, technically it does not require automation. However, automation of software development, testing and deployment through continuous delivery is widely recognized as a key enabler of DevOps. Automation enables organizations to deliver software more quickly while ensuring operations can have confidence in what is being deployed, and customers get the quality, security and stability they require.

The DevOps Trinity

A lens for viewing DevOps culture in a simplified form focuses on DevOps as being about gaining alignment between all software development lifecycle participants on three planes – people, process and tools – or what we can think of as the DevOps trinity.

In practice, misunderstandings on what DevOps is or shortcuts taken in a rush to implement DevOps often result in an organization attempting a DevOps transformation without considering all three components of the trinity. Failure to respect all components almost always leads to failure and unmet expectations.

For example, consider an organization that dismisses cultural change as a requirement. Seeking to accelerate the delivery of higher quality software, this team approaches DevOps as a tools and technology challenge. The organization makes a significant investment in automated tools for testing, but neglects to institute a cultural focus on quality first. Without the cultural focus on quality, teams go through the motions, but don't actively take steps to ensure quality constantly and consistently. Important aspects of quality include setting effective quality goals, implementing the appropriate levels of automation or collaborating across silos to correct problems early and quickly. Without a focus on these aspects of quality, an organization is unlikely to see improvements in quality or reductions in defects. Instead, they find they can deliver more quickly, but at the cost of decreased quality and more firefighting when defects inevitably arise.

Conversely, an organization fully committed to cultural change but unwilling or unable to adopt agile methods and automated tools will find that the cultural change is simply too difficult to sustain, in practice. With manual steps, a heavyweight process and ill-fitting, cumbersome legacy tools still in place, the expectations for faster delivery go unmet and the transformation is eventually abandoned as a failed initiative.

Framing DevOps

So far we've talked about DevOps in terms of a culture based on certain principles and in terms of the trinity of people, process and tools. The truth is that there are several models for defining and describing DevOps, and all of them can lead to a deeper, fuller understanding as well as a smoother DevOps implementation. One good example is the Three Ways of DevOps described by Gene Kim and his co-authors in their insightful book *The Phoenix Project*. Of course, these ways – systems thinking, amplification of feedback loops and a culture of continual experimentation and learning – overlap significantly with the principles outlined earlier. Another good example is the CAMS model, which overlaps similarly with its focus on culture, automation, management and sharing.

Returning to the idea of the trinity, there is one more way to frame DevOps that resonates with many in IT. In this framework, the software development lifecycle is viewed as having upstream (development) and downstream (operations) halves. The two halves are part of the same software delivery process but in many non-DevOps IT organizations these halves are highly disconnected (Figure 1).

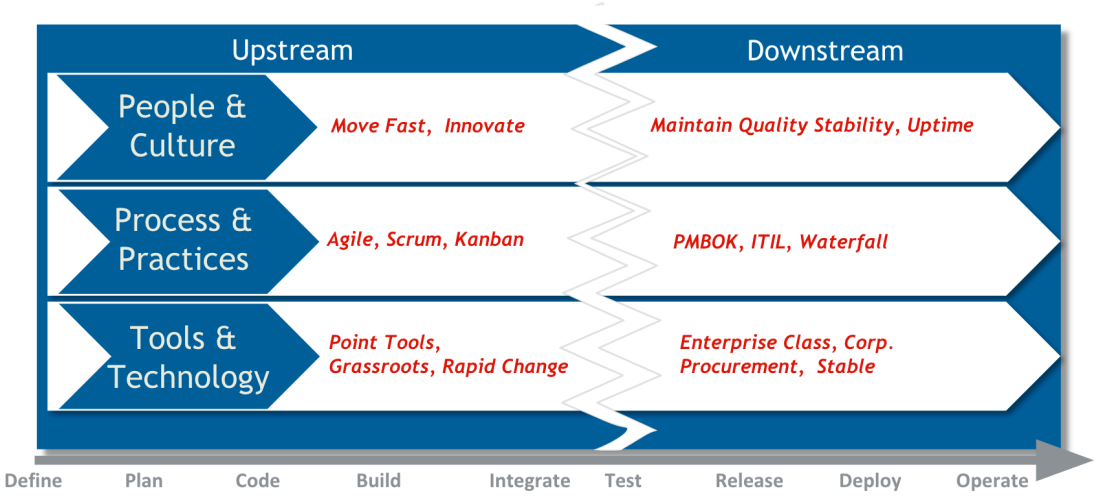


Figure 1: The disconnect between upstream and downstream people, process and tools.

Upstream, the development culture usually prioritizes speed and innovation, whereas downstream the operations culture is tasked with a focus on maintaining quality, stability and uptime. Upstream, development uses point tools to define and build software using agile methods. Downstream, enterprise class tools are the norm for managing the test, release, deployment and operation of the software. Downstream meetings are much more likely to be filled with talk of Information Technology Infrastructure Library (ITIL) and Project Management Body of Knowledge (PMBOK) than Kanban and the latest scrum. DevOps is about connecting these worlds and eliminating the chasm that exists between the upstream and the downstream.

A Look at the "How" of DevOps

With a better understanding of what the DevOps state looks like, the next question for an organization is, "How do we get there?" It's a deep question about which numerous full-length articles and entire books have been written. The steps needed to change a culture are not easily summarized and much depends on the organization making the transition. There are however, a few steps any organization can take to begin paving the way. You cannot "do" DevOps, but you can, to start, do agile development with continuous integration (CI) upstream. Similarly, you can do continuous delivery (CD) downstream. Success with CI and CD depends heavily on automation, because automation not only saves time but it also reduces defects, increases consistency and enables self-service. By automating CI and CD and encouraging open communication and collaboration, organizations begin to span the chasm that separates the upstream from the downstream and establish the foundation for a DevOps transformation.

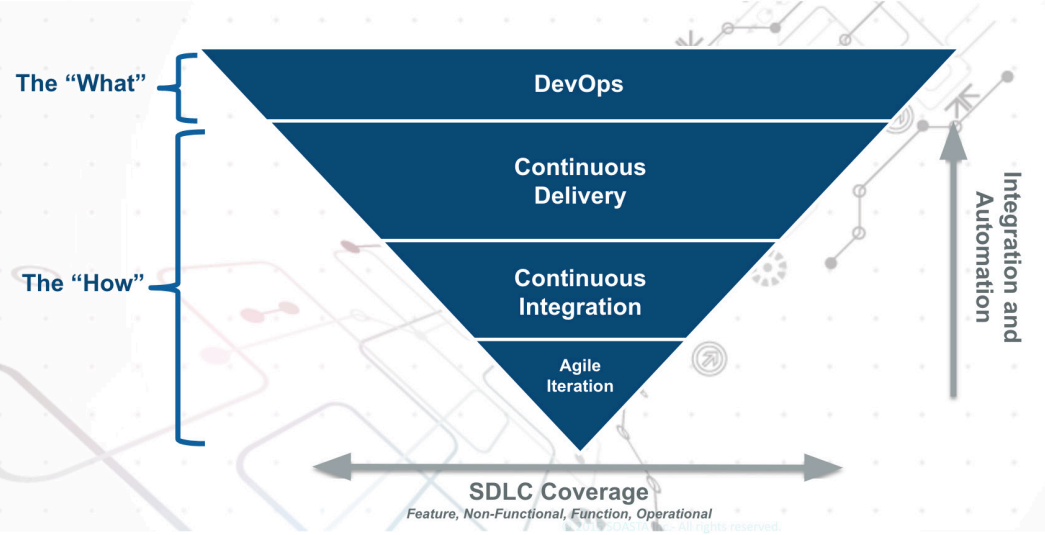


Figure 2: Agile methods, continuous integration and continuous delivery provide the foundation for a DevOps transformation.

Often, discussions on the how of DevOps are too narrowly focused on the technical core of what happens from the time developers commit code to the time software is deployed to a server. But in reality, it's just as important to ensure the customer's needs are understood. Once they are, then you must have a plan to define a solution, a plan to deliver it and a plan to support it once it moves into operation. The entire feedback loop spans more than just Dev and Ops, and that is why it's important to recognize that DevOps extends from concept to customer.

Keep in mind that the benefits of a DevOps transition do not suddenly appear on day one, once an organization has achieved a particular state. Rather the benefits accumulate as the transition gathers momentum. As it grows stronger and more consistent, an organization's ability to accelerate the delivery of higher quality software becomes a competitive differentiator, enabling the organization to innovate, eliminate waste and respond rapidly to market needs.

Why You *Must* Care about DevOps

DevOps is becoming the de facto standard for software development. A quick look at industry examples – think Apple, Netflix, Tesla, Uber, AirBnB and many others – reveals early successes and tremendous potential. Companies that have adopted DevOps principles are disrupting industries, innovating faster and leaving competitors behind. By adopting a DevOps culture, these companies have aligned all stakeholders – from development and operations teams to management and more – around the common objective of delivering quality software rapidly and reliably.

Yet, in the face of mounting evidence of the tangible benefits DevOps provides, many organizations are reluctant to begin a DevOps transformation. Several factors underpin this reluctance, including simple resistance to change. But, as General Eric Shinseki, former chief of staff of the U.S. Army, said, "If you don't like change, you're going to like irrelevance even less."

Another factor is widespread misunderstanding of what DevOps is – a misunderstanding amplified by vendors who say buying their tool will make you a DevOps organization. This confusion contributes to uncertainty about how to adopt DevOps practices.

Yet other organizations dismiss DevOps as a passing fad, or simply don't think that DevOps applies to them because they are not software companies. The facts, however, support the growing consensus that DevOps is here to stay, and that the benefits of DevOps extend to organizations in any industry. If they have not already started, organizations will need to transition to DevOps soon to remain competitive – even to stay relevant. At first, that may sound like hyperbole, but an increasing number of companies are finding out the hard way that almost every company is a software company, that developers will seek employment at companies where they can innovate instead of fight fires all day and that DevOps is a common sense way to gain a sustainable, competitive business advantage.

DevOps Can Disrupt Industries and Speed Innovation

When the healthcare.gov website launched and promptly fell over, there was plenty of blame to go around. Development teams had access to effective tools for software configuration management as well as CI and CD – two enabling technologies for automating DevOps practices. The teams failed, however, to commit wholly to these practices. It took months to respond to the failed launch, and the resulting political fallout led to highvisibility resignations.

Contrast that outcome with that of JPMorgan Chase, which provided its customers with secure, stable mobile apps for traditional banking activities as well as new, modern capabilities such as remote check deposits and instant funds transfers. Providing customers with these capabilities early and rapidly helped Chase improve its ranking from the lowest in customer satisfaction among large banks in 2010 to the highest in 2014[2].

These two examples are representative of the challenges experienced by many organizations that struggle to deliver software and show the kinds of disruptive, accelerating effects that DevOps can have. Companies that have embraced DevOps are better positioned to solidify their position in existing markets, expand into new markets and even disrupt entire industries. DevOps not only helps Netflix deploy thousands of times each day, for example, it also helped the company disrupt the entire cable and TV industry.

DevOps is for Everybody

Today's technically savvy customers are accustomed to having immediate access to the benefits of technological innovation, and they've grown increasingly intolerant of software that is buggy, unstable or not secure. Customer expectations are driving what is known as the Application Economy, in which customers experience a business through its software. In this economy, technology is a business enabler and the ability to reliably deliver quality software is critical.

Even companies in industries far afield from software have found that they need to become software companies to best support their core business. Consider Ford Motor Company: The Ford F-150 truck features more than 150 million lines of code. A lead engineer on the Ford EcoBoost engine team noted that the "secret sauce" in the success of EcoBoost technology is software.

Any company that needs to deliver quality software faster needs to care about DevOps and the supporting practice of CD, which enable continuously building, testing and deploying software in frequent, incremental releases. As Henry Ford once said, "Nothing is particularly hard if you divide it into small jobs."

DevOps is Common Sense

Historically, software was developed using variations of waterfall methodologies with release cycles measured in months or years. Many of the software development practices used today are outdated, having come into existence in the days of shipping software on disks, running batch jobs on a mainframe or even sorting punch cards. In those times errors were costly, thus processes were focused on preventing errors, often by moving slowly through multiple redundant reviews and approvals, rather than finding them quickly and eliminating them.

However, defects inevitably crop up and rework is needed. As such, in following CD practices, release cycles are measured in days, hours or even minutes, and focused on identifying and correcting issues fast. DevOps is a reflection of the industry revisiting outdated concepts and applying common sense changes to meet the demands of the Application Economy.

DevOps Increases Employee Satisfaction

Aside from the software delivery advantages DevOps provides, there are also organizational and human reasons to care about DevOps.

In many software organizations, teams spend too much time fighting fires, moving endlessly from one fire drill to the next. Few people enjoy working this way, and the inefficiencies grind productive work to a crawl.

Most people who have chosen software as a career were attracted by the opportunity to be creative, to innovate and to solve interesting problems. When individuals instead spend their days lurching from one crisis to another, the result is decreased job satisfaction and increased personnel turnover.

Companies that have adopted agile methods and CD practices in support of DevOps have seen an increase in satisfaction for development process stakeholders. Because they are applying modern practices, these companies are attracting and retaining better talent. When employees no longer perform tedious, rote manual tasks and are freed to innovate and make a difference, they are happier. A large financial software company saw a 15 percent improvement in employee satisfaction just weeks after adopting CD in support of DevOps principles. Just as important, a happier and more engaged workforce means increased productivity, lower costs and better software.

DevOps is Not a Passing Fad

It can be tempting to think of DevOps as just another passing fad. A few years ago the buzz was all about software-oriented architectures (SOA), but that buzz faded to a whisper. Even though SOA is an architectural pattern and not a process pattern, it can still be instructive to compare SOA with DevOps. One of the reasons behind SOA's diminished popularity is that it was heavily-architected and ultimately cumbersome. Today, SOA has given ground to microservices, which are more finely grained, smaller and lightweight, making it possible to align the application architecture with smaller, feature-focused, crossfunctional teams. Architecturally, the shift has been from heavier and cumbersome to lighter, agile and flexible. The same shift has occurred in moving from heavy waterfall processes to the agile methods and CD practices underpinning DevOps. Across many industries the momentum is behind more agility and flexibility, making it clear that DevOps is not a fad; it's part of a long-term, sustainable trend.

DevOps is Doable

Making the transition to DevOps may seem difficult, but it is doable by maintaining a focus on pragmatic principles, not dogmatic practices. However, a company should not pursue a transition based on the recommendation of industry analysts. Companies should evaluate the potential difficulties and advantages of a DevOps transition for their specific organization, with an appropriate amount of skepticism for claims that DevOps is the cure for all that ails every software development organization.

In the end, must you care about DevOps? Ultimately, that is for you to decide, but the rapidly mounting evidence strongly suggests that if an organization has not yet begun transitioning to DevOps it had better start soon, before its competitors do and leave it in an untenable market position.

DevOps – Fad or Here to Stay?

In software development, as trends gain more popularity they gain new adopters at ever increasing rates. Often, it is simply because that's what everybody seems to be doing and there is a Fear Of Missing Out (FOMO). This leads many organizations to get swept up in fads, some of which are so short-lived that the fad has passed before any real benefits are recognized. Then on to the next development craze.

Today, DevOps is the trend that is grabbing headlines and attracting all the attention. While some organizations have resisted making a DevOps transition – either due to confusion about what DevOps entails, concerns that it may just be a passing fad or a simple aversion to change – other organizations have jumped in with both feet. The organizations in this second group are eager to reap the benefits being realized by competitors and other companies that have already transitioned to CD and a DevOps culture. These companies are differentiating themselves and delivering higher quality software faster by aligning development and operations across the DevOps trinity – people and culture, process and practice, and tools and technology. Better alignment across these three planes enables organizations to improve time to production, drive business value and reduce IT costs.

According to a Cap Gemini report[3], about 60 percent of the organizations surveyed have already implemented a DevOps approach or are planning to do so in the next 24 months. That leaves 40 percent with no DevOps plans at this time. So, which group has it right? Is it the group that views DevOps with skepticism, having seen too many tech industry fads fizzle before delivering on their promise? Or is it the group that is embracing DevOps and establishing DevOps for the long term? At times like this, it pays to step back, take a breath and take stock of where we are and where we are going. When we do, it becomes clear that DevOps is not a fad; rather it is the way successful organizations are industrializing the delivery of quality software today and will be the new baseline tomorrow and for years to come.

DevOps Today

When research shows that most companies surveyed are planning to or are already transitioning to DevOps enabled by CD, that speaks to the current popularity of DevOps. It does not fully address the question about the status of DevOps as a passing fad or lasting foundational shift. An answer to that question can be found by taking a closer look at the companies that have already adopted CD practices and a DevOps culture. Many of these companies are the established leaders in their respective industries and hold a dominant market position. Having made substantial, long-term investments in CD and DevOps, these companies decided some time ago that DevOps is no mere fad. As a result they are already seeing impressive benefits. For example:

- **Amazon**, the leader in public cloud infrastructure, now releases on average every second[4]
- **Etsy**, the leading peer-to-peer e-commerce marketplace, went from deployments that took hours to deploying changes 50 times per day[5,6,7]
- **Ticketmaster**, the world's leading ticketing company and one of the world's top 10 e-commerce sites, deploys to production at the end of every sprint or multiple times per sprint[8,9]
- **Netflix**, disrupter of the entire cable and TV industry and responsible for about 30% of North American internet traffic, deploys thousands of times daily[10,11]
- **Nordstrom**, a leading fashion specialty retailer with 323 stores in the U.S., increased the pace of releases to monthly from just twice per year[12]
- **Allstate**, the largest publicly held personal lines property and casualty insurer in the U.S. is now delivering new applications in half the time it took to deploy a single new feature[13,14]
- **Coca Cola**, the world's third most valuable brand, has accelerated project delivery as much as 50% and cut defects in production by about half[15]

DevOps Tomorrow

Looking ahead to the next few years, it's not difficult to see where DevOps is heading. Investments in DevOps will begin to yield higher returns. DevOps practices will become better defined, and there will be less confusion across industries about exactly what DevOps is, which will further cement DevOps as a standard in software development. Furthermore, as DevOps becomes more industrialized, we will see DevOps incorporated into formalized training and integrated into university curriculums.

Indeed, as DevOps becomes ingrained in successful companies, those companies will serve as positive examples of the benefits of DevOps to the enterprise.

Companies that effect a successful DevOps transformation are going to retain and attract talented people, who will drive further quality improvements, productivity improvements and ultimately success at the company. Meanwhile, companies that fail to embrace DevOps will bleed talent and fall further behind their competition, serving as illustrations of the adapt-or-perish maxim that is as true in the software development world as it is in the natural world.

DevOps in the Next Five Years and Beyond

Over the next few years, DevOps will reach mainstream acceptance. A leading analyst firm pegs this stage at about four to five years from now, but it is likely to be sooner.

When looking back at similar trends in software, patterns emerge that shed light on the future of DevOps. The name or familiar buzzword of the trend tends to fade, while the underlying tenets and principles are absorbed by organizations as the new standards for delivering software. For example, in 2016 fewer people are talking about Extreme Programming (XP), but the agile development methods and continuous integration principles that support XP are now standard practices for today's successful teams.

In five years, we may not be using the term "DevOps" as much – we may no longer have DevOps groups or DevOps in job titles – but what will remain are the underlying principles of using automation to bridge the gap between development and operations across people and culture, process and practice, and tools and technology.

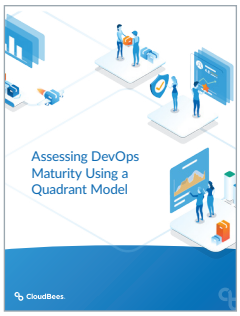
A little more than 100 years ago, the automotive industry was revolutionized by the first moving assembly line. Once industrialization and automation were applied to dramatically reduce the time required to produce an automobile, there simply was no going back. Certainly a few car manufacturers resisted the change, clinging to their existing approach, but the holdouts all eventually adapted or they perished. DevOps is the industrialization and automation of software development, and it is rapidly becoming the norm. There is no going back. As you survey your market today, it is safe to assume that your more successful competitors will be embracing DevOps in five years or sooner. The question you must answer is, "Where will your organization be in five years?"

CloudBees Drive Enterprise DevOps and Continuous Delivery

CloudBees CI delivers Jenkins for the enterprise with a tested and verified Jenkins core, advanced enterprise features that extend the power of Jenkins, fully-tested updates and professional support for the Jenkins core and all 1,400 community plugins. CloudBees CI provides high availability to support continuous delivery processes, as well as simplified manageability, security and scalability of Jenkins to ease administration across distributed Jenkins environments and IT architectures, on-premise or in the cloud.

[1] *TechBeacon, 10 companies killing it at DevOps*, September 25, 2015, http://techbeacon.com/10-companies-killing-it-devops
[2] *JPMorgan Chase & Co. 2014 Annual Shareholders Report*: https://www.jpmorganchase.com/corporate/annual-report/2014/ar-downloads.htm
[3] *Capgemini, DevOps - The Future of Application Lifecycle Automation*, https://www.capgemini.com/wp-content/uploads/2017/07/devops_pov_2015-12-18_final.pdf
[4] *ZDNet, Joe McKendrick, How Amazon handles a new software deployment every second*, March 24, 2015, http://www.zdnet.com/article/how-amazon-handles-a-new-software-deployment-every-second/
[5] *InfoQ, João Miranda, How Etsy Deploys More Than 50 Times a Day*, March 17, 2014, https://www.infoq.com/news/2014/03/etsy-deploy-50-times-a-day
[6] *New Relic blog, Asami Novak, Going to Market Faster: Most Companies Are Deploying Code Weekly, Daily*, February 4, 2016, *or Hourly*, https://blog.newrelic.com/2016/02/04/data-culture-survey-results-faster-deployment/
[7] *TechBeacon, 10 companies killing it at DevOps*, September 25, 2015, http://techbeacon.com/10-companies-killing-it-devops
[8] *tm/TECH, Roshan Revankar, Fear and Paranoia in a DevOps World*, November 14, 2014, https://tech.ticketmaster.com/2014/11/14/fear-and-paranoia-in-a-devops-world/
[9] http://www.livenationentertainment.com/ticketing/
[10] *New Relic Blog, Asami Novak, Going to Market Faster:Most Companies Are Deploying Code Weekly, Daily*, February 4, 2016, *or Hourly* https://blog.newrelic.com/2016/02/04/data-culture-survey-results-faster-deployment/
[11] *TechBeacon, 10 companies killing it at DevOps*, September 25, 2015, http://techbeacon.com/10-companies-killing-it-devops
[12] *TechBeacon, 10 companies killing it at DevOps*, September 25, 2015, http://techbeacon.com/10-companies-killing-it-devops
[13] *Pivotal Cloud Foundry blog, Paul M. Davis, Allstate's Andy Zitney Is Disrupting How Insurance Does Technology*, May 27, 2015, https://content.pivotal.io/blog/allstate-s-andy-zitney-is-disrupting-how-insurance-does-technology
[14] *Allstate website*, https://www.allstate.com/about.aspx
[15] *TechValidate*, Case Study: The Coca-Cola Company, March 19, 2014, https://www.techvalidate.com/product-research/ibm-devops/case-studies/1B0-FCA-9AD

Learn More



Visit the CloudBees DevOps Resource Center
www.cloudbees.com/devops

Read the Whitepaper
Assessing DevOps Maturity Using a Quadrant Model

Read Case Studies and See How Real-World Enterprises Are Using DevOps to Transform Software Delivery
www.cloudbees.com/customers

Learn more

www.cloudbees.com/products

CloudBees CI is built on top of Jenkins, an independent community project. Read more about Jenkins at: www.cloudbees.com/jenkins/about

© 2020 CloudBees, Inc. CloudBees is a registered trademark and CloudBees CI, CloudBees CD, CloudBees Cloud Native CI/CD, CloudBees Feature Flags, CloudBees Build Acceleration and CloudBees CodeShip are trademarks of CloudBees. Other products or brand names may be trademarks or registered trademarks of their respective holders.

0820v03

CloudBees, Inc.
4 North Second Street | Suite 1270
San Jose, CA 95113
United States
www.cloudbees.com
info@cloudbees.com