



YOUR CFO CARES
ABOUT DEVOPS —
THE ECONOMICS BEHIND
CONTINUOUS DELIVERY

Applications are a key driver to the growth and success of a business. Visibility into application performance and the efficiency of the software development lifecycle reaches far beyond IT and well into business stakeholders. Applications are complex and fast-evolving, and with the expansion of mobile and web consumption demanding flexibility to handle constant requirement changes effectively, businesses are forced to reassess their application delivery strategies.

New software development methodologies have emerged to address the need for agility starting from development practices to full automation of the software release process. This collection of best practices have matured into the continuous delivery (CD) process, which applies industrialization concepts to software. Designed to streamline and accelerate software delivery while ensuring that reliable software is released, CD creates an alignment between the application and changing business needs. The principles of CD find their roots in the DevOps movement, established to bridge traditional gaps between software development and the operations team running applications. DevOps automates manual testing and release processes by replacing them with scripted procedures. It extends Agile development in building applications incrementally to include the full integration, testing, and validation phases. By streamlining application delivery across the entire development lifecycle, applications can be iteratively developed, automatically packaged and tested, and then released to production in a rapid, consistent manner. The CD paradigm establishes the notion of an ever evolving production-ready version of the application in a functional, deployable state throughout its lifetime. Adopting CD allows companies to realize direct financial benefits averaging 19 percent in revenue increase¹. Application maintenance costs reduction and efficient utilization of IT staff increase these gains in addition to less tangible benefits such as customer satisfaction, brand reputation, and quality perception.

¹ "TechInsights Report: What Smart Businesses Know about DevOps", CA Technologies, September 2013.

DISRUPTIONS HAVE ALREADY CHANGED SOFTWARE DELIVERY

Cloud technology has commoditized IT environments, creating more freedom of choice for running production applications. On demand, flexible computing capacity changed the economic models for application infrastructure. Cloud allows infrastructure decisions to shift based on cost assessments and real time traffic patterns rather than cumbersome IT dependencies. As a consequence, the production environment becomes less predictable and can change multiple times during the life of an application. At the beginning of 2011 only 20 percent of Zynga's daily active users were running on the company's private cloud while 80 percent were on the public cloud. Just a year later, nearly 80 percent of Zynga's active users were running on their private cloud². Modern applications are becoming infrastructure-agnostic and developers must adjust their processes to support application delivery practices in which the business logic code is portable and can run virtually anywhere. Enabling the organization to take advantage of production flexibility requires the design to anticipate hybrid and changing production environments. Developers cannot rely on a single production target over the lifetime of the application.

New systems of engagement, in particular the vast expansion of mobile and web application access, force companies to react quickly to shifts in end users' behavior and adjust to scale and usage pattern swings. Zynga, for example, grew its pool of servers by as much as 1,000 per week over a two year period in order to accommodate booming traffic driven by social media games². This scale burst was unimaginable prior to the distribution of rich mobile apps and social web consumption. This massive expansion in the number of connected devices is still in full motion. IDC expects the installed base of connected devices will be approximately 212 billion globally by the end of 2020. This will include 30.1 billion installed autonomous connections in 2020³. The ability to develop and deliver high-quality software faster is crucial to companies trying to capitalize on mobile use expansion, creating rapid and massive shifts in application workloads.

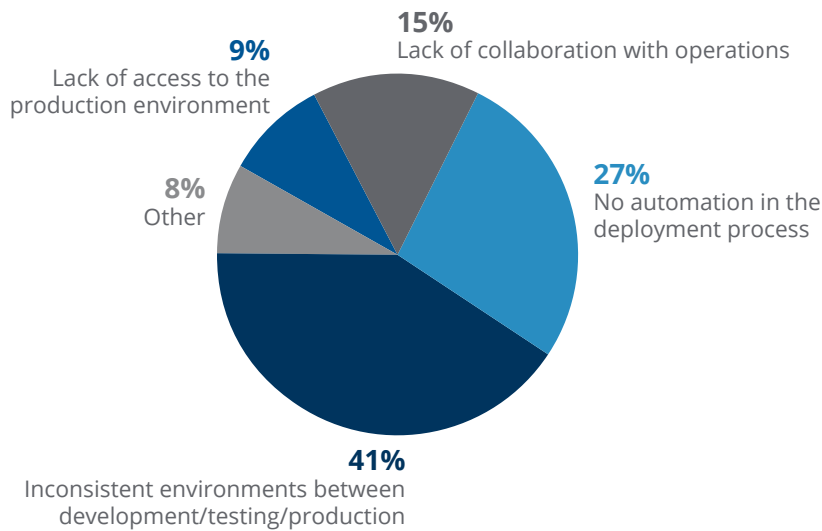
The majority of development projects still haven't reached a level of maturity that can address a rapid delivery mindset. In a survey of more than 4000 developers⁴ commissioned by Rogue Wave Software, 87 percent of respondents reported having experienced delays in moving their applications from development to production. When asked to identify their main challenges, 68 percent pointed to lack of automation and environment inconsistencies as the main cause of release delays while 15 percent pointed out lack of collaboration between development and operations.

² "Zynga Moves 1 Petabyte of Data Daily; Adds 1,000 Servers A Week", Leena Rao, techcrunch.com, September 2010.

³ "The Internet of Things Is Poised to Change Everything, Says IDC", Business Wire, October 2013.

⁴ "Zend Developer Pulse Survey", Zend, a Rogue Wave company, June 2013.

Based on your experience, what is the main cause of failure/delay in rolling out your code to production?



Base: 4,159 enterprise, SMB and independent developers worldwide
Source: Zend Developer Pulse Survey 2013

Early DevOps initiatives focused on accelerating software delivery, and changing the culture to establish better collaboration. DevOps promotes stronger communication skills and gives developers a wider responsibility over the application. Overall the intent is not for a single entity to cover both application development and management, nor to encourage developers and operations to perform each other's duties, but rather smooth the collaboration between these groups. The requirement for more multidisciplinary skills for developers created a new DevOps role combining application development skills, IT operations understanding, and infrastructure automation techniques. IDC's 2014 survey revealed that 43 percent of the Fortune 1000 companies' respondents were already using some DevOps practices. An additional 40 percent were evaluating DevOps methodologies⁵.

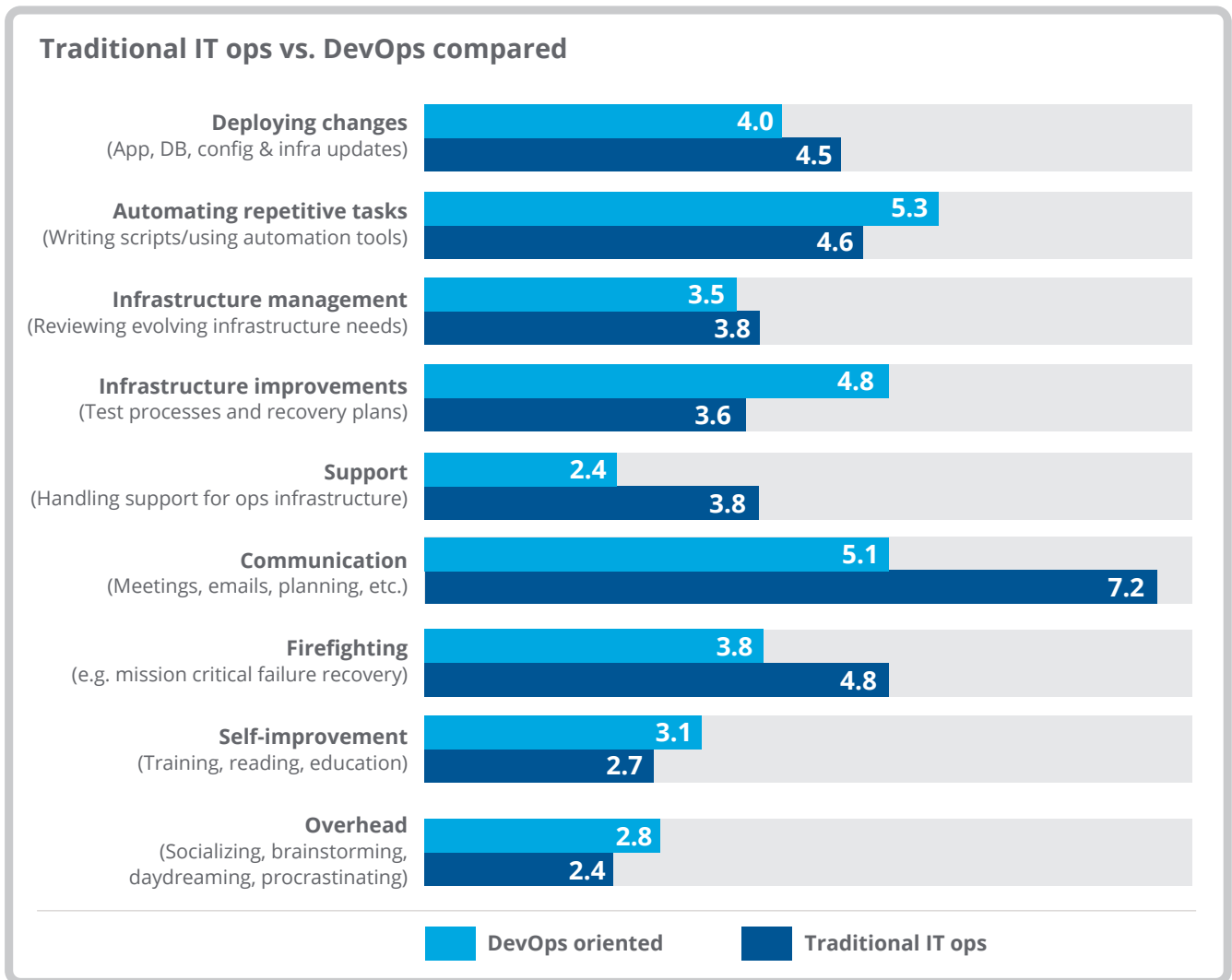
YOUR CFO SHOULD SUPPORT DEVOPS, TOO

The DevOps movement, initiated primarily by developers, laid the groundwork for more comprehensive software delivery automation practices that gradually matured. Software can now be delivered at will rather than at planned intervals, hence CD. Adopting a CD methodology requires not only a change of mindset but a change of tools, standardization on new processes, and implementation of best practices. Integrating the proper tools into the release process automates distinct manual tasks and synchronizing steps in the release cycle. At its best, such a process requires no manual intervention other than quality control and regulatory checks, in essence industrializing software development. Companies implementing CD practices are therefore industrializing software manufacturing by applying an assembly line approach to the software delivery process.

⁵ "IDC's DevOps Best Practice Metrics: Fortune 1000 Survey", IDC, December 2014.

A survey commissioned by CA Technologies reveals an average of 19 percent revenue increase directly attributed to the adoption of DevOps methodologies¹. A drill down to the most common financial attributes of DevOps adoption provides more clarity. Automating the software delivery benefits the organization financially in multiple value areas:

- Revenue gains from enhanced developers productivity and reduction of IT headcount waste
- Revenue gains from accelerated time to market of new functionality
- Gains from cost reduction of application failures resulting from increased quality
- Gains from flexibility in the IT environment



Automation of manual tasks and consistency of the environments, minimizes the amount of repetitive error-prone work done by developers and reduces IT headcount waste. The IT Ops & DevOps productivity report⁶ shows that DevOps-oriented teams consistently invest more in proactive automation and process improvement tasks while saving on average about 4.9 hours per week per person delivering their core IT tasks. DevOps-oriented teams gain an overall 7.2 hours or 16 percent of productivity per week. Developers also spent significant time on problem resolution and

¹ "TechInsights Report: What Smart Businesses Know about DevOps", CA Technologies, September 2013.

⁶ "IT OPS & DEVOPS PRODUCTIVITY REPORT 2013 TOOLS, METHODOLOGIES AND PEOPLE", Rebel Labs/ZeroTurnaround, April 2013.

maintenance versus developing new functionality. The Zend Developer Pulse survey finds that the majority of developers spend 30-40 percent of their time finding and resolving problems and not implementing new functionality⁴. A CD process that incorporates comprehensive diagnostics methodologies, including static and dynamic analysis, significantly reduce the time developers spend on problem resolution. Diagnostic solutions shorten the problem isolation and identification tasks which consume the most time often reducing the overall resolution time by at least 50 percent to a total saving of up to 18 percent in developer's productivity.

Much like an automated assembly line, CD allows the software manufacturing organization to produce new, reliable code on an ongoing basis. Delivering enhancements to address the current market needs and security threats drives direct revenue and provides competitive edge. The business impact of accelerating time to market credited to shorter cycles depends on the industry and is specific to each company, averaging 20 percent and increasing when software drives the majority of the revenue. CA Technologies finds a 20 percent average reduction in time to market and a 21 percent growth in new functionality delivered to the business¹. Flexibility in the software manufacturing process not only ensures faster delivery of revenue generating applications, but it also enables more advanced organizations to adjust their IT infrastructure costs based on changing market conditions and price fluctuations. Companies increase their profitability by scaling their environments up or down based on demand, transitioning between cloud vendors or hybrid environments, and taking advantage of the flexibility in the commoditized IT space.

Quality is one of the major concerns and a usual driver of high costs and project delays. Application errors are expensive and negatively impact organizations in the form of reputation damages, customer loyalty, and direct financial impacts. Improving application quality is often the trigger behind building a DevOps practice. IDC finds that critical failures in production applications at Fortune 1000 companies reach \$500,000 to \$1 million per hour⁷. The State of DevOps survey finds that high performing organizations experience at least 50 percent fewer failures and restore service 12 times faster than their peers⁷. Organizations practicing DevOps demonstrated a mean time to recover of less than an hour in 47 percent of incidents compared to 17 percent in a traditional IT practice⁸. 40 percent of the companies practicing DevOps are able to restore services in under 30 minutes and only 22 percent take 60 minutes or more⁹. Even a very conservative estimate still demonstrates 30-minute resolution time acceleration per incident by adopting DevOps. The National Institute of Standards and Technology indicates that the cost of finding errors in production are 6-30 times higher compared to identifying them during the development and testing cycle. The actual costs of fixing defects discovered in production can become tremendous when the software is embedded and can't be easily updated or when the application processes high-value transactions. Quality improvements and earlier defect detection are bigger adoption drivers in such applications.

CD provides best practices for optimizing each stage in the application delivery process. An automated manufacturing assembly line assumes a reliable, consistent, and high-quality supply chain. To achieve code consistency and quality, development teams adopt Agile software methodologies and mandate the use of diagnostics tools at the individual developer level while automated testing and environment provisioning assure that high-quality standards are maintained all the way to production.

¹ "TechInsights Report: What Smart Businesses Know about DevOps", CA Technologies, September 2013.

⁴ "Zend Developer Pulse Survey", Zend, a Rogue Wave company, June 2013.

⁷ "DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified", IDC, December 2014.

⁸ "2013 State of DevOps Report", Puppet Labs & IT Revolution press, 2013.

⁹ "Zend Developer Pulse Survey", Zend, a Rogue Wave company, 2014.

THE KEY PRINCIPLES OF DEVOPS MATURITY

For DevOps cultural change to mature into effective CD practice, the IT leadership must adopt and embrace a number of key principles.

- **Production readiness** – The fundamental principle behind CD is the ability to deliver a production-ready release on demand. The organization must reach a maturity level in which the application code is always in a production-ready state. Production readiness does not necessarily mean that the application is deployed to production on a continuous basis, but rather the organization retains the ability to do so at will.
- **Uncompromised quality** – Software quality cannot be compromised at any point and the development organization has to prioritize high quality over new features. Ensuring a consistent high quality requires developer's responsibility and proper tooling. It demands tiers of comprehensive testing: Unit testing and static analysis before build and automated functional testing, load, and endurance testing with proper runtime diagnostics tools in place. Quality failures abort the build process until resolution.
- **Repeatable delivery** – The entire software delivery process from build through staging to production must be reliably repeatable so that each iteration is performed in an identical manner. This is achieved by adopting IT tasks automation. Repeated manual tasks that are prone to errors and inconsistencies are also wasting expensive IT resources and time. Automation of these tasks is a prerequisite to any successful CD process.
- **Frequent build and integration** – A CD environment operates with the notion that changes to the application code between build cycles are minimal. Agile, incremental development is practiced alongside CD to ensure that the development project is broken into short iterations. Builds are triggered on all code checked-in to ensure that problems are isolated and addressed quickly.
- **Application stack consistency** – The application stacks should be consistent and automatically provisioned to eliminate environment configuration inconsistencies. Consistency also accelerates the developer's and IT problem resolution capability as it reduces the failures related to application external dependencies.
- **Diagnostics and application management** – High code quality requires problem detection and immediate resolution as defects occur. Fast and meaningful diagnostics data becomes critical to a successful CD implementation. Static analysis and dynamic analysis tools are sequentially deployed during the build cycle providing developers with the insight and drill down data. Lack of developer insight and diagnostics information allows defects to slip through and delay the ability to deliver a quality build.
- **Broad test automation coverage** – Test automation is a prerequisite to ensure high quality and production readiness. Unit tests and multiple layers of automated functional tests are implemented to identify potential issues and regressions. Developers are required to develop unit tests for each submitted piece of code. Automated code quality and unit testing during the integration phase should cover at minimum 75 percent of the code base¹⁰ to establish a reliable process. Automation could run four to five times more tests compared to manual testers¹¹, and should be coupled with diagnostic tools to accelerate resolution.

¹⁰ "Continuous delivery: reliable software releases through build, test, and deployment automation", Jez Humble and David Farley, Addison-Wesley, 2011.

¹¹ "Automated testing ROI: fact or fiction?" Hewlett-Packard Development Company, February 2009.

SO WHERE IS MAINSTREAM ADOPTION?

DevOps practices see a significant rise in adoption across industries, though misconceptions tend to delay implementation of a complete CD process. About 50 percent of developers surveyed for the Zend Developer Pulse report have adopted at least one DevOps-oriented practice in their organizations and about 30 percent were in the process of adopting additional methodologies. However, only 5 percent of developers have adopted the full set of practices required for establishing a CD process. That gap is often a factor of initiatives driven by developers compared to a comprehensive company strategic move. The perceived upfront investment is often considered too high, but moreover, there is a notion that establishing a CD practice is a massive undertaking that interrupts the organization for a considerable length of time. Thirty-five percent of companies that have adopted or are looking to adopt DevOps practices see cost and time as their primary barrier to move forward, while 32 percent find their biggest challenge as lack of knowledge on how to begin the journey¹². Since CD is evolving from a collection of DevOps practices that can be implemented independently, over time and to different degrees of maturity, each of these — continuous integration, test automation, diagnostics, or release automation delivers business value autonomously. It is therefore more common to see companies adopting distinct elements of CD first with less disruption to the organization, and maturing gradually to demonstrate end-to-end automated release capability.

Netflix, Amazon, Facebook, and other industry leaders are publicly known for their ability to deliver new features quickly to production. Facebook, for example, developed principles and a management style that push its developers to continuously innovate and improve. The developers are moving and learning fast even if things get broken in the process^{13, 14}. Facebook released new code to production twice a day⁵, using a dedicated release team and developing tools to ensure that new changes do not cause functionality outages. Amazon established its leadership as an Agile powerhouse releasing code to production hundreds of times each day. Impressive as these capabilities are, most organizations do not need to operate at such production release rates. That said, adoption of CD delivers significant productivity, efficiency, and financial benefits even when the organization doesn't require frequent deployments to achieve its business goals. This is a fundamental difference between continuous deployment that suits primarily vendors providing online services and continuous delivery which provides organizations with the flexibility to release at will. In fact, many enterprise DevOps organizations that implement CD keep sufficient and strict acceptance validation and authorization gates prior to actually pushing an application package to production.

Organizations tend to adopt DevOps over time and gradually progress along the maturity curve. The route to DevOps typically begins with developers' commitment to code quality, version control, and proper unit testing. Learning from organizations that already started the transformation of their software delivery process we can see that 60 percent of DevOps-oriented organizations have adopted continuous integration and application diagnostics solutions according to the Developer Pulse Survey. These are common practices to adopt first. Infrastructure standardization and comprehensive test automation are practiced at about 30 percent of these organizations¹¹ and bring the organizations closer to a complete process. Eighty-five percent of developers that already implemented some DevOps practices have realized the benefits and based on their initial gains, they intend to embrace a more complete process over time.

⁵ "IDC's DevOps Best Practice Metrics: Fortune 1000 Survey", IDC, December 2014.

¹¹ "Automated testing ROI: fact or fiction?" Hewlett-Packard Development Company, February 2009.

¹² "Zend Developer Pulse Survey", Zend, a Rogue Wave company, August 2012.

¹³ "Facebook's Zuckerberg explains 'the Hacker Way'", Benny Evangelista, SFGate, February 2012.

¹⁴ "Mark Zuckerberg's letter to investors: 'The Hacker Way'", wired.com, February 2012.



Rogue Wave Software 1315 West Century Drive, Suite 150 Louisville, CO 80027
Tel 800-487-3217 · Fax 303-473-9137

© 2016 Rogue Wave Software, Inc. Zend and Zend Server are registered trademarks of Rogue Wave Software, Inc.
All other trademarks are the property of their respective owners.