UK Edition
Chapter Extract

Tech Trends 2014

Real-time Dev-ops

Download the full report at **www.deloitte.co.uk/techtrends**

Deloitte
University
Press

Real-time DevOps

Empowering the business of IT

IT organisations need to better respond to business needs with speed and agility. IT can likely improve the quality of its products and services by standardising and automating environment, build, release, and configuration management – using tools like deployment managers, virtualisation, continuous integration servers, and automated build verification testing. Popular in the agile world, DevOps capabilities are growing in many IT organisations with either waterfall or agile methodologies.

WHEN it comes to application development (Dev), the business cares about speed and quality. How fast can I get what I want? How close will it be to what I need? Contrast that with IT operations (Ops), held accountable for response times, stability, and efficiency, and focused on how to reduce business disruptions at the lowest cost. These are very different core missions that yield very different behaviours. Dev is looking to compress delivery cycles and adopt "experiment and learn" mentalities. Ops is looking to institute controls and more tightly govern change. The fact that the "build" and "run" shops are typically separate organisations only adds to the divide.

Further complicating matters, both Dev and Ops could each benefit from investments in enabling technology – creating automated capabilities in the business of IT akin to what finance, manufacturing, and the supply chain have invested in over the past decades.[1] But even as requirements management, system maintenance, and other disciplines are upgraded, incremental investments in disconnected activities will go only so far.

The real goal is to bridge the gap between development and operations, supercharging the investments that currently exist in siloed automation by integrating the end-to-end delivery model. Simply stated: *Real-time DevOps*.

Some companies have been using automation to accelerate and improve steps in their development processes for a while. But many more have not. In fact, a recent survey of 1,300 senior IT decision makers revealed that only 39 percent had already invested in pieces of DevOps.[2] Complacent about their time-tested, over-the-wall approaches to software development, many IT organisations have settled for manual, disjointed processes that have rightly earned a reputation for ineffectiveness and inefficiency.

It's complicated

Complexity surrounds DevOps. From basic design principles and defect tracking to release management, configuration management, and more, the interdependencies of the software development lifecycle are real and complicated. All the more reason to automate and integrate the process.

The rise of agile is one of the factors driving increased interest in real-time DevOps. What was once seen as an

The Dev

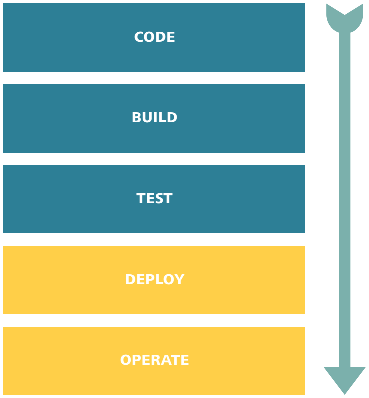
Innovates and creates applications.

The Ops

Keeps infrastructure running smoothly.

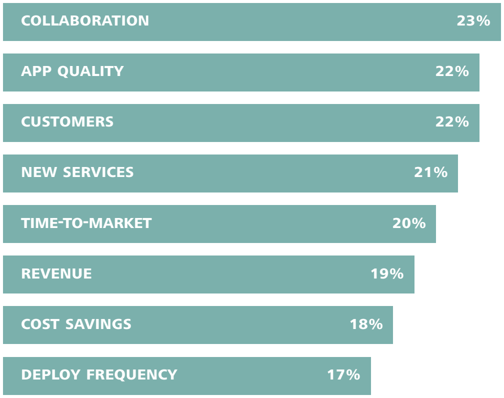
DevOps disciplines

Integrate and automate processes.



The benefits of DevOps[1]

Percent improvement in business areas.



Sources: [1] Computer Associates, "TechInsights report: What smart businesses know about DevOps," https://www.ca.com/us/register/forms/collateral/techinsights-report-what-smart-businesses-know-about-devops.aspx, accessed January 3, 2014.

experiment is now mainstream, with more than 70 percent of companies reporting at least partial agile adoption.[3] At its core, agile is about "test, experiment, learn, and repeat." It's based on short sprints – typically one or two weeks – to get to a potentially releasable product. That's instead of development cycles that last many months or years.

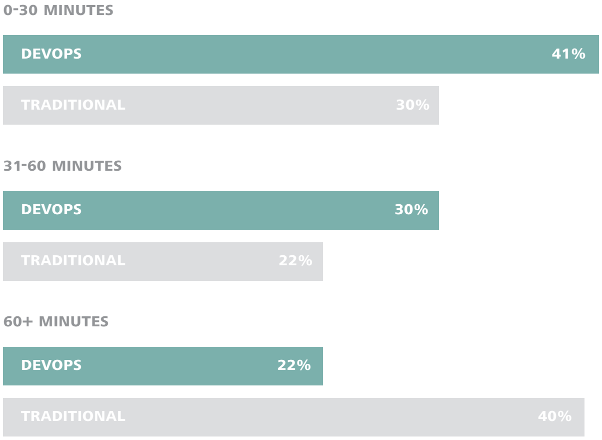
Accelerated time to market is one possible benefit, but just as important are the side effects of better managing changing priorities and improved alignment with the business. Agile backlogs represent potential work to be completed – stemming from new ideas, unmet requirements, or enhancements and fixes coming back from operations. Each sprint sees the business sponsor (or product owner, in agile parlance) re-establish priorities. Development teams then tackle those items on the top of the list.

Contrary to popular misconceptions, agile development often requires a more disciplined approach than traditional software development methods. It can also provide more transparency in the development process – and real-time visibility into progress. Iterative, rapid development begs for structure and rigour, which are available through the automated, integrated capabilities of real-time DevOps.

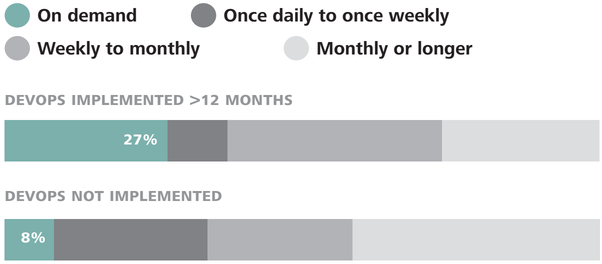
Continuous build and continuous integration are big parts of real-time DevOps. Code is constantly being fed back into configuration management and validated via testing automation suites, giving developers affirmation that quality is continuously monitored, with dependencies managed automatically in the background. Companies gain the ability to automate technical compliance and structural correctness, as well as a continuous measure of how requirements are being met.

Environment provisioning and management is another opportunity area. Traditionally, projects had to accept up-front capital spend and ramp-up delays while hardware and software were ordered, installed, and configured. With the rise of virtualisation, cloud, and software-defined data centers, this process can be largely automated: not only

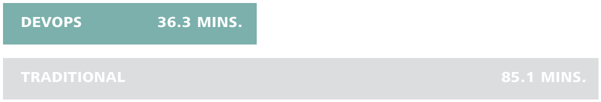
Production failure recovery time[2]



Deployment frequency[3]



Deployment duration[2]



[2] ZeroTurnaround, "Rebel Labs IT ops & DevOps productivity report 2013," April 9, 2013, http://zeroturnaround.com/rebellabs/rebel-labs-release-it-ops-devops-productivity-report-2013, accessed January 6, 2014. [3] Puppet Labs, "2013 DevOps report,"March 2013, http://info.puppetlabs.com/2013-state-of-devops-report.html, accessed January 6, 2014.

procurement, but also configuration of servers, networks, storage, operating systems, platforms, applications, transactions, and data – including the automated provisioning of user accounts based on profile types.

Real-time DevOps also includes test automation. When coupled with requirements management, functional, user, and even behavioural scenarios can be scripted for automated quality assurance (QA). This benefits development cycles and also serves as the benchmark for regression scripts – accelerating break/fix and maintenance processes. For areas such as web and mobile development, this might also include build-verification services across devices and operating systems – analysing across an increasingly fragmented landscape.

Where there's a waterfall, there's a way

While agile is helping to showcase the need for real-time DevOps, these concepts are just as relevant for waterfall shops. Handoffs and manual steps typically waste time in the development process. Collapsing the time it takes to develop software, and creating more thoughtful linkages to operations, are universal benefits.

Also, real-time DevOps does *not* mean that existing Information Technology Infrastructure Library (ITIL) and governance processes need to be scuttled. Indeed, IT service management should have a more explicit link to software development across the lifecycle, with identified issues fed into the backlog for prioritisation. Ongoing patching and infrastructure upkeep should still be done, and with real-time DevOps, it will likely be better coordinated with development. Improving outage recovery and minimising release fail rates are expected outcomes from real-time DevOps – as well as expedited code deployments and improved developer throughput.

Real-time DevOps is not a tool, though tools make it workable. And it's not only about agile, though agile practices have brought the benefits to light. Instead, real-time DevOps is a process shift that changes the cadence of how much can be done – and in how much time.

Moving at the speed of commerce

John Lewis PLC, one of the UK's leading retailers, replaced its customer facing e-commerce platform in early 2013. This was a complex project involving a team of over 100 employees and consultants working across multiple systems: web storefront, web management, product management, and delivery management. To support this project, multiple development environments were used. Each one was carefully managed to support its respective development stream. Code was then deployed across many additional environments: system testing, integration, performance, training, and ultimately production.

To meet the project's pace and flexibility demands, John Lewis focused on DevOps and took measures to increase the frequency and richness of communications between the development and project operations teams, resulting in a prioritised list of DevOps-related enhancements. Many of these took the form of process automation in order to improve reliability, repeatability, and speed.

Since the go-live in early 2013, the company has continued to develop and refine its DevOps processes, focusing on efficiency and reliability. Automation has continued to be a main theme: Automated browser-based functional tests have been adapted so that they can be used on larger, fully integrated environments as both smoke and regression test suites. With real-time DevOps practices in place, John Lewis can now deliver one complete (back and front-end) release per month. Previously, releases were only carried out every six to eight weeks.

With the positive results from using real-time DevOps practices on the e-commerce project, John Lewis is now expanding these practices to other projects, scaling the operation and creating a clear delivery and project operations team for the enterprise. Additionally, the company is looking to orchestrate its automated processes, enabling an end-to-end, "one click" deployment across multiple systems.

Where do you start?

While there are many opportunities to make a shift to real-time DevOps, there are some places where you likely can't live without it: mobile, social, and big data. In these fast-growing spaces, disjointed, bottleneck-ridden development processes can undermine your efforts. Unless you can find a way to accelerate without sacrificing quality – and real-time DevOps is likely that way – you'll find yourself out of the loop as the business bypasses IT by going directly to the marketplace.

- **Establish the need.** Conduct your own benchmarking to identify delays and waste in the software development process. Uncover how much time is spent on manual document capture, build management, build verification, release planning, and test script development. These are opportunities for action.

- **Build new skills.** Tool configuration and scripting skills are a part of the equation, used to drive version control, configuration management, test harnesses, ticketing, environment provisioning, and system maintenance and monitoring. But soft skills are just as essential for real-time DevOps to take hold. Team members will be collaborating with the business, programme and project managers, developers, testers, and the operations teams. Make sure your core team isn't simply making the new technologies adhere to how things have historically been done.

- **Employ services thinking.** For real-time DevOps to be viable with legacy ERP and large-scale custom solutions, break down complex systems into components and modular services. This allows for rapid incremental changes within monolithic code bases – and sets up the organisation for a broader modernisation play.

- **Lay down the bases.** Once you understand the pain points within your organisation, begin automating individual components. Establish a continuous integration build server for your developers, create a small "smoke test" suite of cases to validate builds, and implement a release automation tool. Also, look to add automation within your development, test, and infrastructure tracks in parallel with similar, discrete steps.

- **Connect the dots.** Once you have some of the automation components in place, look for ways to link them into a single stream that can shorten cycles. Not just integration between requirements, but continuous integration – linking to build, to defect tracking, to configuration management, and to release management. In this model, the handoffs and touch points happen seamlessly in the background.

- **Get vendors on board.** The opportunity to learn from and build on vendor successes in real-time DevOps is an important way to accelerate your own improvements. You may want to avoid outsourcing agreements with vendors who aren't using automation as an accelerator.

- **Make the leap to test-driven design – or even behaviour-driven design.** Real-time DevOps enables you to move from build-to-run to build-to-verify. This natural evolution leads to design for end-user engagement. That's where contractors and vendors that provide application development and maintenance (ADM) and/or application management services (AMS) can help you improve.

- **Look beyond cost and speed.** Embrace the lower costs and greater speed that come with real-time DevOps, but recognise that more substantial benefits are also possible. If you believe that your technology delivery model can benefit from real-time DevOps, it's time to get teams delivering against your priorities with a much more compressed cadence.
- **Commit.** Too many companies dabble in this world – acquiring tools and adopting some of the terminology, but without making hard changes to their operating and delivery models. If there is a case for real-time DevOps, don't fall for one-off, surface-level investments. Or if you do, don't be surprised if you get unremarkable results.

Bottom line

Individual tools for automating the software development lifecycle, maintenance, and monitoring have been available for years, and many companies have been taking advantage of them effectively. Yet few have taken the next step to integrate the pieces and commit to a new cadence of development and operations. That's because the concept of real-time DevOps is only partially understood. In a recent survey, Gartner "found that only one-third of companies surveyed were either in-process or planning to implement DevOps, and close to 44 percent of respondents were still trying to figure out what DevOps means."[4]

Early adopters have the opportunity to profoundly improve their IT shops – accelerating IT delivery, improving quality, and better aligning with their businesses. Arm IT with the tools to automate and integrate their core disciplines, and the cobbler's children will finally have new shoes.

Authors

Ayan Chatterjee, principal, Deloitte Consulting LLP
Alejandro Danylyszyn, principal, Deloitte Consulting LLP

[1] Deloitte Consulting LLP, *Tech Trends 2013: Elements of postdigital*, 2013, chapter 10.

[2] Computer Associates, "TechInsights report: What smart businesses know about DevOps," https://www.ca.com/us/register/forms/collateral/techinsights-report-what-smart-businesses-know-about-devops.aspx, accessed January 3, 2014.

[3] Lori MacVittie, "IT cause and effect: Applications are money," *InformationWeek,* August 13, 2013, http://www.informationweek.com/it-leadership/it-cause-and-effect-applications-are-money/d/d-id/1111158?, accessed January 3, 2014.

[4] Laurie F. Wurster et al., "Emerging Technology Analysis: DevOps a culture shift, not a technology," Gartner, Inc., August 8, 2013.

Contacts

Sam Roddick
Partner
Deloitte MCS Limited
020 7007 8283
sroddick@deloitte.co.uk

Kevin Walsh
Head of Technology Consulting
Deloitte MCS Limited
020 7303 7059
kwalsh@deloitte.co.uk

About Deloitte University Press
Deloitte University Press publishes original articles, reports and periodicals that provide insights for businesses, the public sector and NGOs. Our goal is to draw upon research and experience from throughout our professional services organisation, and that of coauthors inacademia and business, to advance the conversation on a broad spectrum of topics of interest to executives and government leaders.

Deloitte University Press is an imprint of Deloitte Development LLC.

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited (“DTTL”), a UK private company limited by guarantee, and its network of member firms, each of which is a legally separate and independent entity. Please see www.deloitte.co.uk/about for a detailed description of the legal structure of DTTL and its member firms.

Deloitte MCS Limited is a subsidiary of Deloitte LLP, the United Kingdom member firm of DTTL.

This publication has been written in general terms and therefore cannot be relied on to cover specific situations; application of the principles set out will depend upon the particular circumstances involved and we recommend that you obtain professional advice before acting or refraining from acting on any of the contents of this publication. Deloitte MCS Limited would be pleased to advise readers on how to apply the principles set out in this publication to their specific circumstances. Deloitte MCS Limited accepts no duty of care or liability for any loss occasioned to any person acting or refraining from action as a result of any material in this publication.

Registered office: Hill House, 1 Little New Street, London EC4A 3TR, United Kingdom. Registered in England No 3311052.

Designed and produced by The Creative Studio at Deloitte, London. 32430A