DevOps for Digital Transformation: Hit the Accelerator

Gunnar Menzel, Vice President, Chief Architect, Cloud Infrastructure Services

No doubt about it: the future belongs to the fast. Virtually every enterprise is looking to increase the cadence and quality of its business processes—to innovate faster, to get new products to market sooner, to adopt new and more productive ways of working, to accelerate customer service. They're looking to become agile, digital enterprises in order to stay ahead, to increase market share, and introduce new products. That is why two-thirds of CEOs are now focused on digital transformation strategies, according to IDC.

The adoption of DevOps parallels the rise of digital transformation. Almost 75% of technology professionals across a broad cross-section of industries are now using DevOps techniques, according to a recent study by RightScale. The need for speed is a key driver of DevOps adoption: In the era of digital everything, software is the source of differentiation, and DevOps is the key to accelerating the development and delivery of high-quality software.

Yet many companies have not fully realized that there is a direct connection between speeding up DevOps initiatives and expediting the digital transformation journey.

Simply put, if you can't hit the accelerator on software development, testing, change management, and release cycles, your entire transformation agenda will quickly get stuck in the mud. It will bog down in a morass of siloed systems and tools that don't work together, manual processes that take too long, and disconnected teams and people who don't—or cannot--collaborate effectively.

So the new question is how to accelerate DevOps in order to expedite the business outcomes of digital transformation. Here are three key suggestions as a starting point.

1. Approach DevOps from a business perspective.

All too often DevOps is seen as a solution to a technical challenge: for example, too many software changes are creating too many outages, or too many incompatible tools are slowing down the release cycle. DevOps should never be viewed as a solution in itself. You can't buy DevOps by the kilo or implement DevOps as a "project." It's a means to an end; it's a process; and the objective should be viewed in terms of business outcomes, not tools and technologies.

If you see DevOps as a way to increase automation, for instance, you will make very different decisions about how to implement DevOps than if you'd focused on integrating toolsets. On the other hand, if you see DevOps as a way to accelerate innovation, or to increase market capitalization, or to achieve competitive differentiation through near-instant

software change-and-deployment cycles, you will see the big picture perspective of what has to happen from a people, process, and technology perspective.

And if you see DevOps as a way to accelerate your digital transformation, you will make better decisions about how to structure your DevOps journey.

Yes, the right tools implemented the right way can solve all kinds of technical challenges, including a dramatic reduction in the number of change-related software outages. But a top-down focus on business outcomes, rather than a bottom-up focus on IT tools, will more likely deliver the results you're expecting from your DevOps initiative—and your digital transformation.

2. Automate everything you should.

Automation allows IT systems to do what they do best, executing repetitive tasks much faster and more efficiently than any human can perform. IT systems are best at performing repetitive, multi-step processes a thousand per second with each stepbeing executed in exactly the same way and with the same speed and precision

For example, consider moving/promoting code to the live environment. It involves risks and may give rise to outages. With higher manual intervention, the risks of errors and outages multiply. Many times errors aren't caught instantly and may have an impact on other processes; for instance, testing an application change. In this case, the tester would most probably assume that the error is due to the change and would request the developer to fix the issue. The developer would typically use a development environment to write and "unit test" the change. Since the developer's environment can sometimes differs from the test environment, no error is reported, i.e. "it works for me."

By automating builds and integration of code changes, valuable resources are no longer spent diagnosing why the change works in development environment but not in the test environment. Instead, the team can focus on higher-value, more innovative tasks.

Similarly, by automating testing processes, such as unit, functional, performance, and security testing, you can unburden highly talented, highly paid professionals from dull, repetitive, manual testing and give them time to explore new ways of working—and working together—that are more efficient and creative.

And by automating delivery (release management and deployment processes), you can free DevOps teams to work together across platforms and operating systems, keep pace with critical updates, understand and execute on customer requirements more effectively, and iterate changes and refinements faster and with higher quality than ever.

3. Get the feedback you need for continuous improvement.

Constant improvements to both applications and to the DevOps process itself are important goals, and this requires a feedback loop that accelerates the flow of feedback in all directions—from ideas into code and production, from production and operations back to application teams, and from business leadership to DevOps teams. The feedback loop should account for many different variables, including (to name just a few):

- Bug reporting: Providing an easy means to capture (through social media for external sites), triage, track, or add to developers' backlogs
- Feature suggestions: Allowing users to provide their feedback on a service which may come through tools such as UserVoice, which can then be integrated with the lifecycle
- Availability, performance, and usage monitoring: Allowing feedback on operational aspects to provide insights (or maybe report defects) into future developments
- Instrumentation: Allowing more specific measures to be captured which are meant to be used as insights into future developments
- Business use cases: Creating better collaboration by keeping DevOps teams in the loop about what the business is trying to accomplish and keeping the business informed as to how those goals impact users and software delivery

DevOps. Digital transformation. Make the connection.

In a business climate that's all about speed and driven by software, DevOps is the key to superior business outcomes. If you really want to accelerate the business results of digital transformation, start by accelerating your journey to DevOps. Stop listening to vendors who offer specific tools that are intended to solve specific technical issues. Get solid guidance from a technology-agnostic partner with DevOps expertise, an intensely business-focused approach, and local resources on a global scale. Like Capgemini. Talk to us about your goals—and we'll help you create a DevOps strategy that accelerates your digital transformation.

Capgemini

For further information please contact:

infra.global@capgemini.com

About Capgemini

A global leader in consulting, technology services and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2016 global revenues of EUR 12.5 billion.

Visit us at

www.capgemini.com