Busting 5 DevOps Myths

Section 31, Lecture 175

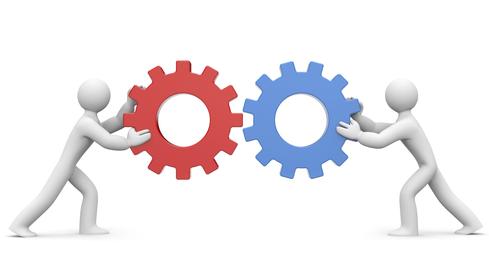
**DevOps is a technology movement for hipsters and your company doesn't need to worry about it, right? Right?**

DevOps, with its focus on combining development, testing, deployment, and operations, has fast become the poster-child for accelerated app delivery. Some will argue that DevOps is long overdue since business success clearly hinges on quickly delivering innovations based on high-quality software services.

**DevOps is for hipsters**  
A good friend of mine is a gun developer. He can write anything from C++ to JavaScript to Rails. He's also 56 with a receding hairline. Like me, he started in IT operations, worked with virtualization years before it became mainstream, and still remembers assembler programming.

Yet if we believe a lot of what we read about DevOps, the hand-off problem where developers throw suspect code over the proverbial wall to operations (which then breaks) can only be resolved by a new breed of blended professional: a "silo busting" individual tooled up with everything from automated configuration to production code simulation.

**My advice:** Before scouring for DevOps talent, review the untapped skills of existing resources in the context of your goals. Even though my friend never talks about DevOps, he epitomizes the movement -- that is, he's an ops professional who thinks like a coder and a coder who understands the need for resilience in everything he develops. However, he isn't the problem. Rather, it's the rigid culture and "one skill in one box" organizational structure that prevents his rounded set of skills from being fully utilized.



**One movement to rule them all**  
It amazes me how folks drawn to DevOps belittle any set of best practices already established. Suddenly bodies of work like ITIL, COBIT and balanced score card fall from favor, with so called DevOps proponents proclaiming they no longer have a place and should be discarded.

**My advice:** Don't get caught up in the hype. You invest in any set of best practices to accelerate and improve the services you deliver and support. While DevOps is grounded in Agile thinking, change, and continuous delivery, the IT service management processes needed to ensure resilience and stability remain more critical than ever. So don't just look at where they compete, but how they can complement each other.

**It's a technology movement**  
There's a lot of really good technical material written on DevOps, with many new terms to consider -- like infrastructure-as-code and antifragile -- all supported by a new range of products and techniques. But while it's valuable fodder, what's often forgotten is the old adage -- automating bad process just results in a faster bad process. So just being the fastest app dev shop in the west with great new tools means nothing if the finished work doesn’t meet business or customer expectations.

**My advice:** Look at what tools you have and how they're being used. Chances are something like application performance management is controlled by the operations team. But what if the developers could effectively use it to instrument code, simulate performance behaviour and better determine capacity requirements before production? Now it's not just a tool for monitoring, it's a collaborative way to reduce defects and improve quality.

Remember also to focus on both speed and "nimbleness" -- so examine capabilities that help you quickly assemble viable applications using customer feedback to test assumptions. With this in mind, accept that finished applications might be completely different than originally intended, so review and be prepared to change methods for project planning, product management, funding, and implementation.

**Our business is immune**  
Many organizations argue that the principles of DevOps don't apply because they've outsourced operations or don't have an application development function. Others surmise that because they're in the product making business or government service delivery, any movement based on driving continuous change has no place in their "ain't broke, don't fix it" world.

**My advice:** Look at business first and IT second. Nike isn't in business because they only manufacture sneakers -- they actually sell a fitness experience (wearable devices). Withings doesn’t sell bathroom scales anymore -- they sell smart body analyzers with an ecosystem of externally developed apps.

Irrespective of infrastructure platform or development methodology, these companies recognize they're in the software business. They use software to create new business models, reinforce their brands, and deliver innovative new services**.**

**DevOps will change the world**  
With so much hype, it’s easy to start drinking the DevOps cool-aid. But consider this -- regardless of best practice, methodology or movement, the success rates of application development projects haven’t improved much in 20 years. Although 2012 marked a high watermark for project success, 61% of projects were still challenged or failed completely.

**My advice:** As a colleague of mine perfectly put it -- DevOps is about five things, people, people, people, people, oh and people. So before jumping feet first into DevOps, review the culture, processes and metrics supporting your teams. For instance, if your operations team has been historically rewarded for stopping application releases then something is definitely wrong and no amount of Agile development will help. Similarly, DevOps won't help you much if your developers won't leave the office to meet customers.