DEVOPS DIRTY LITTLE SECRET: SUCCESS DEPENDS COMPLETELY ON GETTING OPS RIGHT

In his opening remarks at DOES18, Gene Kim acknowledged that the event was "really great for Developers, not as good for Operations". A lead analyst at a major market analysis firm stated in a recent inquiry that "Developers don't think about Ops much".

Look around the DevOps world and the indicators are everywhere. It seems pretty clear that the view from the Dev perspective is that the only reason systems fail or that there are any bugs in code is because Ops are incompetent, ignorant, officious and resistant to change. The solution is AIOps or NoOps or CloudOps or anything that will allow developers full and unbridled access without interference from the nefarious Ops police.

But wait just a minute! The only way ANY business value accrues from applications is once the code is "in production". And if Dev doesn't want to think about operating their applications, who should?

Before you dismiss me as an Ops bigot, let me just state that I am not advocating for any specific solution for operating applications; as long as there is one! The FANGs (Facebook, Amazon, Netflix and Google) of the world care very deeply about operations. They may have differing solutions, such as Google's Site Reliability Engineering (SRE) approach but it is acknowledged that SOMEONE must be responsible for ensuring operational durability. And that consideration should be baked in from the very inception of the application and its design and build-out of the environment in which it runs.

Let's examine some aspects of the DevOps phenomenon. All of a sudden, organizations came to realize the "traditional method" for deploying to production was a huge bottleneck to agility and digital transformation. But why did that situation arise in the first place? There was a time, I know because I lived through it, when there were no barriers to deploying to production.

Developers/engineers ran the entire show. Then as computing became more and more critical to commerce or service delivery, things changed. I would like to argue a contributing reason for the change was the failure by developers and engineers to take sufficient care of production. They deployed code and changes that caused problems and they failed to develop sufficient fallback/recovery procedures, causing outages that impacted the business. And ironically, considering what we hear today, the BUSINESS said no to changes without change boards, implementation plans, back out plans, tests, reviews, hoops and flips. We “evolved” from immediate, almost real-time change implementation to submitting service requests to think about submitting service requests. That was the good old pendulum swinging to one side and now it seems it is, or already has, swung back to the other extreme.

We know a pendulum cannot remain at one side of it oscillation. So, what should be done to stabilize the pendulum in the middle instead of swinging back towards the opposite extreme yet again? I think we should remember George Santayana who is credited with writing "Those who cannot remember the past are condemned to repeat it." Production is where the value is. No matter how cutting edge or cool the application is, you must plan for operationalizing that code in the hostile world of competing applications, demanding users, clever hackers and random chaos that is production. The goal of technology professionals who embrace DevOps should be not to deploy to production but rather to run and thrive there. To accomplish THAT goal, operations must be top-of-mind throughout the SDLC and not just an afterthought at the end.

Learn how a Jobs-as-Code approach to application workflow orchestration can help you accelerate delivery and improve application quality.

Come and check us out! Planning to attend the DevOps Days Zurich conference next month? Be sure and drop by our booth to say hi!