



DEVOPS FOR THE ENTERPRISE: TEAM STRUCTURING AND POSITIONING

👤 Patrick McClory 🕒 June 30, 2016 📁 DevOps 👁 7,099 Views

Historically, teams designing software for release into the production environment have had limited access to the production environment where real-world issues occur. While this model works, it's not necessarily ideal; there are two critical problems with it:

1. Disparate teams are unable to effectively coordinate solutions for regulatory or compliance requirements that span the development and release environments
2. Operation policies ultimately restrict optimal application design

The shift in moving to a DevOps-oriented view of systems development and management requires that teams align and manage according to the quality of the product they are responsible for rather than a specific technical or functional discipline. Historically, software design and production environments are owned by two disparate teams. The concept of DevOps (as simple as the portmanteau suggests) is to remove the separation between Developers and Operations. This gives developers the responsibility for the

operational system and gives operations teams the ability to influence and work within the development lifecycle.

Often this has occurred due to the interpretation of regulatory or compliance requirements that were otherwise difficult to achieve through implementing different processes such as utilizing credential tokenization techniques and targeted, detailed access logs for root cause analysis purposes. With the commoditization of resources across the IT industry, this is no longer the case, especially when leveraging Amazon's shared responsibility model, which provides a base level of security and compliance for the cloud and a number of tools for users to ensure the security of their data within the cloud, like Amazon Inspector, AWS Key Management Services, AWS Identity and Access Management (IAM), and Amazon's Cloud Compliance enablers. Technology, in this case, has matured to serve the use case far more directly and is far easier to manage, configure, and design than ever before.

Along with the divide in the management of teams, the decision making process related to how best manage a given application becomes the responsibility of the team who owns the production environment rather than the team responsible for the design and development. This is not to say that an enterprise-wide standard should not exist, but since the success and quality of the production environment also falls to the same team, they need to be able to adopt processes and workflows that leverage the advantages of and mask the potential deficiencies of the application's architecture. DevOps teams working on customer-facing (either internal or external) applications must also take ownership of support issues in the overall context of the workflow. Ideally, by streamlining the process of releasing updates to the production environment, fixes to end user-impacting problems would take less time and effort.

Managing the shift in processes might seem complicated, but when implementing this type of strategy, there is a strong opportunity to consolidate a standard set of tools (such as task/issue management, automation toolsets and test frameworks) that can serve as the base process that teams start with when designing their specific processes. As with other teams, those that manage this core architecture and toolset must take ownership of the tools and support the DevOps teams that utilize their tools and services. Enabling and facilitating teams working within a DevOps process requires a minimal set of tools that will likely sound familiar, though their usage within an enterprise might expand or change to meet the needs of the individual teams. We'll dive into more information on specific toolsets further in this blog series.

As I've said before, DevOps is not just a realignment of teams. It is a cultural shift. To be successful you have to be able to take advantage of this shift. Change is hard and is something that people generally struggle with. You need to find tangible ways to help your teams connect with the value of the transformation on a

business, technical and personal level. Whether you focus on the benefit to the individual's growth, the benefit to the company's (and possibly their own) bottom line, or connecting the journey to the reasons they pursued a career in technology in the first place, getting individuals to connect to the mission at a deep level is critical.

␣