

What does it take to make network operations more agile?

Jim Metzler

I have to admit that like so many analysts and reporters, I focus a lot of attention on software defined networking (SDN) and Network Functions Virtualization (NFV). It's hard not to give that these two emerging architectures hold the promise to lower capital and operating costs; help IT organizations develop and implement new services more quickly; provide just-in-time capacity to meet dynamic changes in user demand for services; and deliver a higher-quality experience for the end customer. However, the process of implementing new architectures such as SDN and NFV brings up a critical question: How will all of this impact how we operate networks?

As mentioned, one of the primary reasons why IT organizations are interested in SDN and NFV is because they believe these new architectures will enable them to reduce the time it takes to deploy new services. For example, SDN and NFV hold the promise of enabling IT organizations to dynamically spin up virtual network functions (VNFs) such as optimization and security and when appropriate, either move that functionality to a different server or spin it down. However, just because the deployment of network functionality is becoming more agile doesn't necessarily mean that IT organizations will be able to reduce the time it takes them to deploy new services or perform key functions such as root cause analysis. To achieve that goal, IT organizations will also have to make their network operations (NetOps) more agile.

Making network operations agile is a significant challenge. Many IT organizations, however, have a starting point to work with because in order to increase the agility of the application development life cycle a lot of IT organizations have already begun to adopt DevOps. The primary characteristics of DevOps include: collaboration; continuous development, integration and delivery; continuous testing and monitoring; and automation.

In order to increase the agility of NetOps to where it needs to be, it is necessary to adopt the previously mentioned primary characteristics of DevOps, but that isn't sufficient. That follows because DevOps is generally applied to discreet services that are frequently delivered over the Web on a best effort basis. The network environment is different than that and as a result the VNFs envisioned by SDN and NFV create challenges that are not addressed by DevOps. One such challenge is that unlike what happens when delivering an application over the Web, NetOps will need to support dynamic and automated management of service performance and SLAs. This can only be achieved by a policy model that supports end-to-end SLA targets. Another difference is that since VNFs such as optimization and security are chained together to create an end-to-end service, this creates strong dependencies between the VNFs. For example, if a service provider updates an optimization VNF they need to ensure that it is fully compatible with the security VNF(s). As a result much stronger version control and compatibility testing is needed than would be typical for enterprise applications.

In terms of the question raised by the title of this blog, I don't believe that NetOps will become more agile without a viable plan to make it more agile. Hence, one of the key ingredients in

making NetOps more agile is management commitment to set a direction, create a plan and allocate the resources to implement the plan. The good news is that IT organizations are not starting with a blank slate as many of them have already adopted DevOps. Applying the key characteristics of DevOps to NetOps is a good start. However, as explained in this blog, the network environment is different than the application environment and those differences must be factored into the plan that management creates for the evolution of the NetOps function.