LISA ’11 Theme— “DevOps: New Challenges, Proven Values”

THOMAS A. LIMONCELLI AND DOUG HUGHES

LISA ’11 Co-Chairs
lisa11chairs@usenix.org

The theme for LISA ’11 is “DevOps: New Challenges, Proven Values.” DevOps is “an umbrella concept that refers to anything that improves the interaction between development and operations” [1]. While usually associated with Web operations, the tools and techniques are now being mainstreamed into the enterprise. Although DevOps is new, it embodies themes long popular at LISA: automation, performance, scaling, collaboration, and cooperation.

There is an important shift happening in system administration, and we felt it was important to acknowledge this change by making it this year’s theme. What is this change? In the status quo, sysadmins obtain software from vendors and struggle with operational issues with varying levels of support. Scaling, disaster recovery, operational efficiency is left as an exercise for the user.

Recently there has been a trend towards self-sourcing. A company providing a Web-based service develops it in-house and the operations work is done by in-house system administrators. Development and Operations work together, with shared responsibility for the success of the whole. This last part bears repeating: shared responsibility.

Let’s call this new way “DevOps.” While we’re at it, let’s acknowledge that it isn’t new. However, it is becoming more frequent and is the dominant paradigm for the high-growth segments of our industry. As a result, new cultural “best practices” are becoming apparent.

While in-house software development is nothing new, the Web-centric world creates opportunities that hadn’t existed before. Likewise, you can manage a new-style, Web-based service with the paradigm of the past, but you would miss out on opportunities that were unavailable before.

For example, the old way often involves packaged, shrink-wrapped software. A new release comes along each year. The effort to ship a new release is huge. You have a printer that makes boxes, a factory that produces media, an assembly line that puts them all together and ships them. Every new release involves a new manual, a new box, and inventory strategies to deal with the old version sitting in the warehouse. All these pieces come together once a year. To make it all happen we use software development methodologies with names like “Waterfall,” “Spiral,” and “Release Trains.” If there are millions of users, there are millions of deployments.

In the Web world, shipping a new release has much lower overhead. There is no physical package. There is, ostensibly, one deployment. This gives birth to frequent releases: weekly, daily, maybe continuously. This, in turn, leads to new software development methodologies like "Agile Development." Yes, you absolutely can use a waterfall model and only update the Web site's software every year, but you would miss out on opportunities that were unavailable before.

Opportunities Arise for Operational Improvements

As mentioned previously, in the old way, sysadmins are solely responsible for operational issues with some or little support from the vendor. Certainly a Web site can be managed that way by treating the in-house developers as the vendor. We can do even better.

DevOps promotes a different culture: developers and operations work together as partners—as a team. The way a manager can bring this about is to make the two groups share responsibility for the operational success of the service.

Previous to DevOps, I'd make feature requests that would benefit the operational efficiency of a service (i.e., make my life easier), and countless times I've seen those feature requests ignored. That attitude changes when the on-call rotation is shared among the developers and the system administrators. Nothing develops empathy for the importance of operational efficiency like a week of pager duty. When informed by operational experience, software development changes in ways that directly benefit the operational efficacy of a company.

It is quite refreshing to leave the "toss each release over the wall" world and enter the DevOps "we're all on the same boat" model. Collaboration between developers and system administrators means that the operational aspects of each new feature are worked out ahead of time. Rather than developing every feature a deployment may need, teams can focus on just the operational features needed by your deployment. Developers have a better appreciation for what information should be logged to ease debugging and what variables need to be exposed to do proper monitoring and metrics.

We're All Programmers Now

In such an environment, system administrators need to become more like developers. Automation becomes critical. Sysadmins have always been "pro-automation" but "who has time to automate anything?" is such a frequent refrain that outsiders would think some of us are anti-automation. There may be justifiable reasons to not automate something, but three of them are disappearing:

1. You can't automate physical work such as installing a new machine.
2. It doesn't make sense to automate something that happens once or rarely.
3. Management isn't funding automation projects, because they don't see the value.

The first objection disappears when using EC2 or other "infrastructure as a service" (IaaS) cloud providers. When installing a new machine is an API call, we're all programmers now.

The second objection disappears because nothing happens once anymore. With old-style packaged software, once it is installed it is installed. The next upgrade might be a year from now. In the Web world, scaling makes very few things "rare." A one-in-a-million error happens hourly and becomes worth fixing. Requests previously done manually must be turned into "self-service" portals so that there is less waiting. If developers frequently need a server's OS reloaded, why should they wait for a system administrator to do that? The portal can verify they own the machine and do the entire process. If developers need another machine, why should they wait for a system administrator to purchase, install, and configure it? The portal can allocate a virtual machine and bill the developers' project code.

The third objection disappears because, in a Web environment, management does see the value, or at least good management does. Velocity becomes important. Uptime becomes important. And, even more importantly, operational efficiency becomes a competitive advantage. These things require the consistency and scale that only automation can achieve.

DevOps reflects a cultural change that reflects the new paradigm. DevOps is a culture. It isn't a job description: you can't hire a "devop." It isn't a technology: you can't buy a software package that provides the "devops service." It isn't a job title: people do not have "devops" on their business card.

DevOps Beyond the Web

DevOps is mature enough that the innovations are now feeding into areas outside Web-based services. LISA is a unique opportunity to apply the lessons of DevOps to traditional enterprise computing, storage administration, security, and network administration.

Traditional computing organizations need the new insights that DevOps culture brings. The stellar uptime of Google, Facebook, and other popular Internet sites has created high expectations for the most simple internal Web app. People want to be able to fill out their expense report forms anytime, even nights and weekends. That was easy when doing so meant a paper form, since paper has incredible uptime. Now such forms are online and we sysadmins are under pressure to make sure they are always available. Packaged software may still ship yearly, but security updates are a constantly flood comparable to the launch schedule of Web sites.

This "mainstreaming" of DevOps is important to us as an industry. It is LISA's great responsibility, as the leader in advancing the state of the art in system administration, to make this happen. DevOps and USENIX LISA embody the same cultural values: automation, performance, scaling, collaboration, and cooperation. These are the values we've always seen at the LISA conference since it began 25 years ago [2].

In a recent phone conversation, Andrew Hume asked Tom to define DevOps. After Tom rambled on for five minutes, Andrew interrupted, "Oh, so they've given a name to the way I've been doing things for years!" He wasn't that far off. Tom reviewed all the presentations from LISA '10 and determined that 27% could easily be classified as "DevOps" and 31% could be classified as "mostly DevOps." Thus it is easy to assert that last year's theme was DevOps but we didn't know it. If we can achieve similar ratios in 2011, the theme will be a success.

LISA '11 will include many new speakers, as well as many familiar faces.

We look forward to seeing your familiar face there too!

References

[1] This definition is attributed to John Allspaw.

[2] By the way, this is the 25th LISA. Happy Silver Anniversary!