CODE

The Secret Of DevOps: It's Always Been About People, Not Technology

ADAM JACOB · JUL 29, 2015

An early proponent of DevOps speaks out.

Adam Jacob is a cofounder and CTO of Chef, an IT-automation company that helps enterprises use DevOps to build and deliver quality software quickly.

I have had the privilege of being either physically present or near many of the seminal moments in the evolution of DevOps. My friend and cofounder at Chef, Jesse Robbins, was the conference chair for the operations track at O'Reilly's Velocity conference.

It was there in 2009 that John Allspaw and Paul Hammond gave their seminal talk, "10 Deploys Per Day: Dev And Ops Cooperation At Flickr." In it they outlined all the ways they worked together to deliver code quickly and safely.

John and Paul’s talk is all the more relevant today. The largest companies in the world are working as hard as they can to figure out how to become better at building and delivering software. It's not an optional piece of their strategy: It's the future of how their customers want to work with them. Software is table stakes for survival.

Work Style Is What Matters, Not Tools

So where should these companies look to understand how to become better at software delivery? Many are in the midst of this transition, but none have completed it entirely.

In some industries, we can point to market-leading disruptors, such as Amazon in retail. But the real answer lies in looking at what John, Paul, and Jesse were doing in 2009—not in the specific technical choices they made, but the style in which they worked, the essence of what they believed made them high functioning and successful.

Then the challenge begins: how to apply this new style to businesses trying to become better at building and delivering software.

See also: DevOps: The Future Of DIY IT?

This is the DevOps movement. The issues that led to DevOps in Web-native companies are now being felt by more traditional big companies. So today there is a noisy market in teaching people how to do DevOps, or in tools that enable DevOps. We'll start seeing certifications, definitions, books, and trainings. Vendor after vendor will tell you they have the magical solution to this difficult business problem.

At its heart, though—when you dig down underneath the rhetoric and positions, the software architecture decisions and design patterns, down below the software-development lifecycle and your agile practice—DevOps is about bringing together all the people you need to build and run your business effectively, and empowering them to move as quickly as possible towards their goals.

Tools matter. Make no mistake, trying to change the way you work without changing the mechanisms by which you do that work is a futile exercise in excruciating failure. But tools exist in service of the prime directive: building highly functioning, highly effective cross-functional teams, that attack your thorniest business problems as a unit, rather than as lone individuals or silos with competing incentives.

See also: Three Reasons Your Startup Will Suffer Without DevOps

Fundamentally DevOps is about taking the behaviors and beliefs that draw us together as people, combining them with a deep understanding of our customers' needs, and using that knowledge to ship better products to our customers. This is how I define DevOps:

> *DevOps is a cultural and professional movement, focused on how we build and operate high velocity organizations, born from the experiences of its practitioners.*

Notice that I don't mention technology, which may be surprising coming from the CTO of a DevOps-focused company.

DevOps is cultural in that it encompasses the ideas, customs, and behavior of a group of people. It is professional because people can often make a living practicing these ideas in a variety of professions—I've seen everyone from CEOs and lawyers to software developers and sales reps practicing DevOps. It covers both the building and operating of our organizations.

It is uniquely suited to, and born from, the attempt to move at ever higher velocity (speed with a direction—it's not just that we move fast, it's that we move fast and we know where to go).

Finally, it comes from experience. It wasn't a business-school theory that got applied to the world, or a mathematical formula that got applied to a problem—it comes from the collected experience of thousands of people learning the hard way how to function in these environments.

What DevOps Is Made Of

From here we can break DevOps down into three different components:

1. the core principles by which it operates, which are shared by every person and organization that does DevOps
2. the forms of work we believe reinforce those principles (such as blameless post-mortems, continuous integration, or iterative versus incremental software development)
3. the application of those principles and forms into our daily work

Each of these components gets progressively more distinct to the individual. We're very similar in our core principles, we share many of (but not all) the same forms, but we're often quite different in how we apply the same to our actual work. This perspective provides a good place to start: understand the core principles first, then start doing some of the things that people who are DevOps experts do, then see how it feels in your own environment.

There is more background on DevOps and other core principles, but in context of this story on the culture of DevOps, the following principle is most relevant:

People Over Products Over Companies

Anyone who has worked in enterprise IT in the last 20 years has likely experienced the inverse of this principle—an environment where we believed more in companies than products, and products more than people. We used to talk about being an "IBM shop," or a "Microsoft shop," or a "CA shop." When you had a problem, the first answer was always to look at the company whose solutions you had de facto decided to adopt, see if they have a product that fits, and if so, use it. Regardless of whether it was the best product, or if it resonated with the people who were going to have to implement and use it day to day.

DevOps flips this on its head. The first question we ask is what the people who are doing the work, or who are going to consume it, need. If we can find a product that fits that need, great, let’s use it. If we can't, we're not afraid to build it ourselves—to take our fate in our own hands and simply solve our problem.

If we consistently find products that solve our problem from the same company, that's great. We value those relationships, too. But the moment it stops providing value, we're taking our money (and our people) elsewhere.

This principle also holds in how you design your internal teams and systems. Sad people build sad products, which in turn create sad companies. Help your people be awesome, and they will make awesome products for you. Make awesome products, and you have a shot at making an awesome company.

DevOps is about people more than technology. The DevOps movement is the solution to the challenges facing every large company in the world right now. Learning its history or principles is valuable—but in order to truly understand what to do next, you need to understand deeply the day-to-day behaviors of the people who practice it.

And in turn, you must learn to apply those in your own context, whether you’re a software developer or a sales leader. Then we use technology to reinforce those behaviors, and as an accelerant to the changes we need to make.

Photo by Travis Isaacs