



DEVOPS – WHAT'S THE POINT? (HINT: THERE IS ONE!)

👤 Patrick McClory 🕒 January 26, 2016 📁 Automation 👁 3,375 Views

Earlier this year, DualSpark merged with Datapipe to form a singularly unique team that cannot only talk the talk, but also walk the walk when it comes to cloud-native strategy and DevOps engineering capabilities. As we continue to move forward, pairing up our deep strategic thinking with our engineering muscles, I've come to realize that it's very easy to seek out the latest in popular trends for technology without actually understanding how or why it may help you. DevOps is one such phenomenon, where it's both pervasive amongst the popular media and press but elusive from both a definition and the ability to achieve this hallowed 'DevOps' status. I'm not going to add color to either of these two sides of the puzzle but want to ask the more basic question – why bother?

Honestly, if what you have going for you works in today's day and age, why change it? What I will say is that when you're in an industry that is seeing more traffic and traction from hungry startups, the large enterprises start to realize that what they were doing served a purpose. It got them to where they are today, but as the technology maturity curve continues to accelerate, they have some work to do to be able to leverage the tools of today to achieve business outcomes at a velocity that meets the needs of their customers.

Having spent time with hundreds of clients – large and small – over the years, I've found that there are two major sets of players in the area of adopting a DevOps practice or strategy: the culture warriors and those looking for new weapons (tools) in the fight against inefficiency. Let's be honest. They're both great ideas,

but I think that they both miss the mark. Winning the culture over and reimagining how businesses can collaborate is a great idea, but it really only speaks to the mechanical linkages and flexibility of the organization. I also both have strong opinions as well as great examples of where tooling has been a major boon to organizations who are in transit from an old way of doing things to a new one. Realistically, tooling is only a part of the 3-legged stool of people, process, and technology that organizations are forced to balance on a daily basis.

To clarify, businesses don't generally (nor do they need to) be entirely altruistic in their path forward as most commercial organizations are interested in serving customers external to the organization while also making a profit. Conversely, companies always have to balance the needs of their team, culture, and processes with the functions of the business. Tools are a cost that must be balanced with the benefits that are received—and cost comes in many forms from licensing to hardware (or instance time on cloud) and support (paid or self-administered). Teams purchasing tools without understanding the balancing act between cost and benefit are less successful in serving the business.

So if the point is clear, then I'm left with one more question: How do I measure it? There are a few measures that our teams have found to be successful:

Mean time to resolution can be defined as the measure of time from when a critical issue is logged or identified in a production system to when it's fully resolved in the same system. From a business perspective, the ability to fix things when they aren't efficient is critical for everything, from maintaining customer satisfaction to meeting contractual obligations. This isn't a new metric, but is critical from the perspective of understanding how quickly things can be fixed if and when they go wrong (and they certainly do from time to time). Using every means possible to tighten the loop from a developer's desk to the production environment means further automation gets injected into the process to speed up deployment, including testing to maintain the integrity of the deployment process.

If you take the mean time to resolution definition and turn it a bit on its side, you get a similar measure that identifies the amount of time required to go from an idea being prioritized to when it's deployed in production. This obviously is focused less on break/fix scenarios and more on a new feature or function workflow, but it also includes an entirely different set of up-front processes. Time to market measures how long it takes for your team(s) to design the solution as well. For this measure, it's critical to not only understand the big picture of how long it takes from end-to-end, but also each of the phases involved including design, prototyping, development, test validation, and deployment. This is a great way to identify

which areas of your process are lagging or where there’s opportunity to dig more deeply into how tooling and culture change can help the business competitively and deliver value to customers faster.

Development velocity is actually one of my favorite measurements for this process and it’s basically the measure of how many or how much business value engineering teams deliver in a given time period versus how much time they spend on non-differentiating or non-business value tasking. Let me be clear: I’m NOT saying that those non-differentiating or non-business value tasks aren’t important, but what I have found is that, in nearly every case, teams who spend more time on these tasks create more duplicative work (or at least exert duplicative effort) in these areas than anywhere else. Whether it’s measured via the velocity of burndown of story points from an Agile/SCRUM or it’s measured against a new feature release, development velocity is a metric that is useful in demonstrating direct progress against business objectives where there’s a healthy prioritization process in place for the engineering team(s).

Culture shift and the right tooling to enable DevOps as a practice within your organization are definitely important, but both of those serve the needs of the business and need to be aligned to a clear goal.

Understanding your operations requirements and how technology aligns with the goals of the processes can ensure your automation plans benefit your organization as a whole.

»