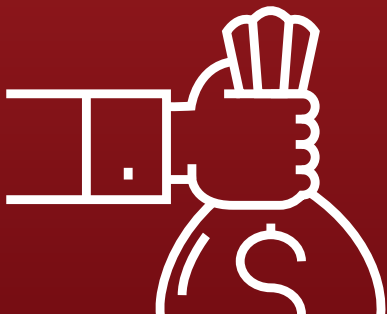




# 10 TIPS

**FOR ACHIEVING CULTURAL  
CHANGE IN DEVOPS**



**An explosion of mostly open source tools and cloud-native platforms has helped to break down many barriers to software development, following the advent of DevOps as a concept more than 10 years ago. This has also allowed organizations to put into practice the mantra that “all companies are software companies.”**

But channeling all of the tools and processes available in such a way throughout the entire development and deployment cycle of software development is contingent on developers, operations, security staffers and other stakeholders taking part in the process throughout the production cycle. In other words, a successful DevOps depends on marshalling people for this to happen, which usually involves convincing employees and entire teams to change how they work. And doing that hinges exclusively on the adoption of the requisite culture.

For a large and established organization, this might mean not only completely changing direction of the boat but also replacing the sails, as peoples' job skills and tasks merge between development, operations and security.

The nimbler startup might have instituted a working DevOps culture from the get-go. But since any DevOps can be improved upon, the startup might need to trim its sails as winds change and cultural changes must be made.

Consequently, successful DevOps is like a wheel, with culture representing the hub connected by processes and tools as the spokes. And implementing this culture falls entirely on the shoulders of the people within the organization to make the changes.

The task of convincing employees within the organization to embrace DevOps is certainly never easy. But for your organization to change course, here are 10 ways help people and teams make the shift.

August 2019

# Taking DevOps Monitoring to the Next Level: The 5 Step Guide to Monitoring Nirvana

DOWNLOAD THE E-BOOK

*"We use InfluxData company-wide to gather operational metrics for the business. It is fast, requires minimal space, and has a rich retention policy capability."*

*Lisa Guo, Developer Manager, Nordstrom*



# Everyone must become a vested stakeholder in business outcomes.

DevOps is not about developers checking boxes in a list of their “responsibilities” before the code is processed by another team. Indeed, it has become almost a cliché to describe how DevOps involves eliminating “silos” separating teams and stakeholders; changing culture runs much deeper than removing silos between departments.

Even the most talented software engineer, for example, will waste their talents by focusing on a single application development or tweak before uploading to Git with little concern about security and how operations will be able to integrate the code. Conversely, operations should not run blockage for application development, especially if the application was created to address a specific business need. The cornerstone of DevOps culture must be that every stakeholder has adopted a product management-like mindset to deliver business-oriented results.

Indeed, DevOps teams will only adopt shared responsibility if everyone is in charge of a product, said Jeff Keyes, director of product marketing at Plutora.

“The whole point of DevOps is bringing Dev and Ops together. So, if you want to accomplish that union and create a product-oriented team, we have to change our mindsets to focus on delivering customer value,” Keyes said. “There’s been such lip service about shared responsibility, but [that occurs] only after you’ve become a product-oriented team and you have cross-functional members who are all responsible for the basic stages of ideating, building, managing risk and dependencies and all the things that are associated with release and operations. Until you have a single team that does that, you’re never going to truly have a value-stream approach.”



# DevOps-related performance metrics must be based on delivering value to the customer.



Everyone has a boss and every boss should rely on objective performance metrics to gauge performance. But unfortunately for many—if not most—organizations, the underlying culture is such that DevOps team members' performance is measured by metrics with little or no direct connection to delivering business value. The developer, for example, may care less about security if they are not responsible for unhappy customers when a post-deployment vulnerability is discovered. As such, the developer must be judged by more than just time-to-deployment metrics. And operations team performance metrics do not mean as much if they force the developer team to rely on YAML against their will—it may help operations improve their post-deployment metrics, but it also stifles developers' freedom to create something for end users that no one else offers.

For developers, many metrics in use today, while interesting, have about as much to do with delivering value to the customers as does the number of keystrokes the developer presses during a given time period, Keyes said.

"If you hold people accountable to providing business value, looking at business results such as customer satisfaction, customer adoption and revenue produced, then things change. When that happens, the people in charge of the product have a different focus," Keyes said. "They must press forward, they must iterate, they must improve delivery. And those measurements will become very, very important to make sure the right things happen."

We have all obviously heard about the importance of “team building” ad nauseum. But in DevOps, we are still learning how to build teams to make use of an arguable overabundance of tools and processes to create and deploy software. And the agendas of those with more-targeted know-how and responsibility may be seen as hostile to others with different job roles.

Most of us have been there: Whatever role you have in any organization, your hard work is replaced by something inferior. In the software development pipeline, delays too often occur when someone from security or a supervisory committee requests unnecessary changes to the code, thus delaying a release.

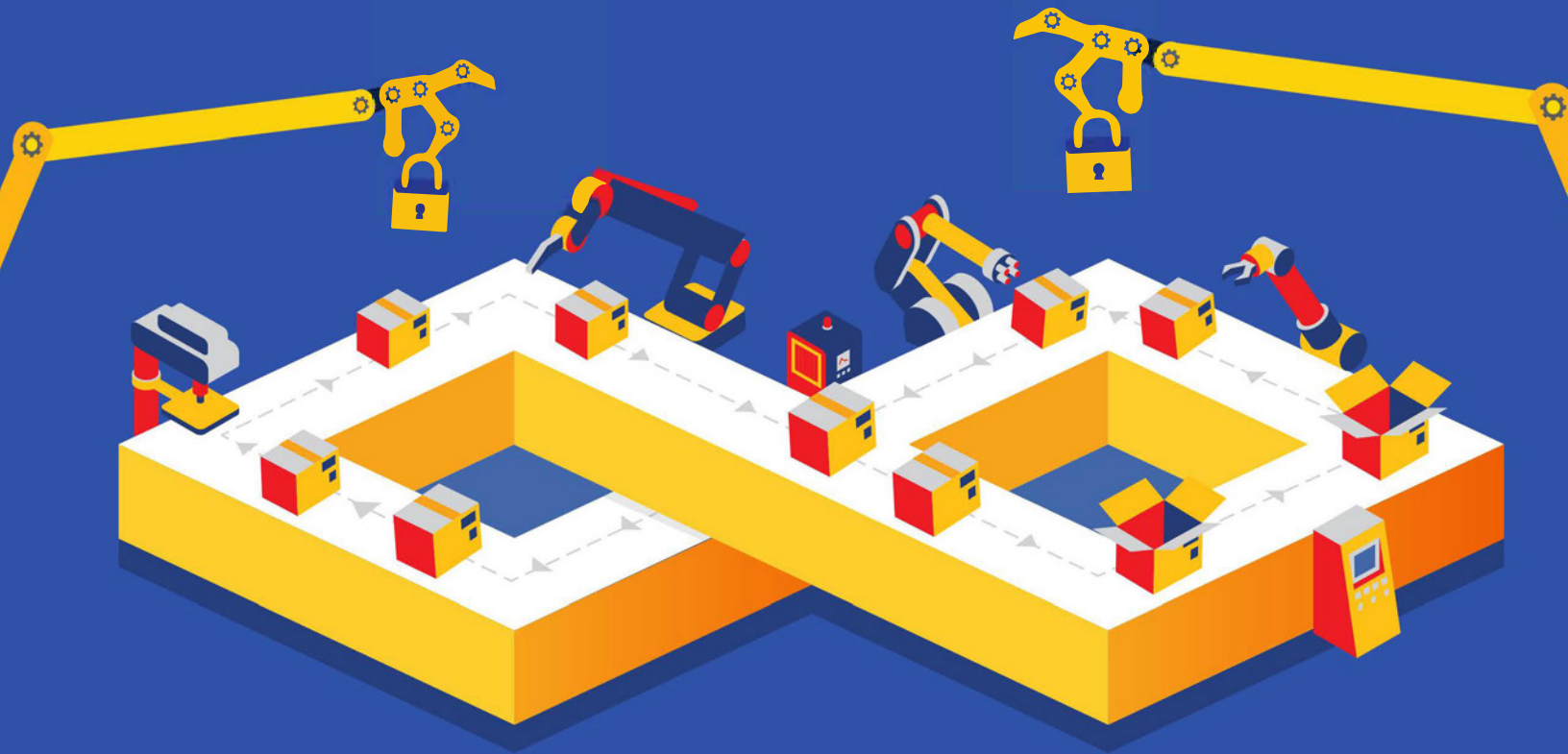
## Everybody must be on the same side.

But it does not have to be that way. While conflicts still often exist between DevOps, DevSecOps, “traditional IT people” and other stakeholders, adopting the right culture—in addition to the right tool and processes—will go a long way to making the cultural shift, Shiri Arad Ivtsan, a product manager for WhiteSource, said.

“Even among our customers, for example, we continue to see a lot of friction between security, as well as the legal teams, the developers and other stakeholders,” Ivtsan said. “Here is where it is critical to first define the right processes, and then make sure all stakeholders have the right attitude to facilitate a DevOps idea and culture. For example, some DevOps have security teams that have the tools to understand the security risks, but they cannot use them properly without the development and other DevOps teams being onboard.”



# Automate Security Into Your DevOps Pipeline



**FIND OUT HOW**

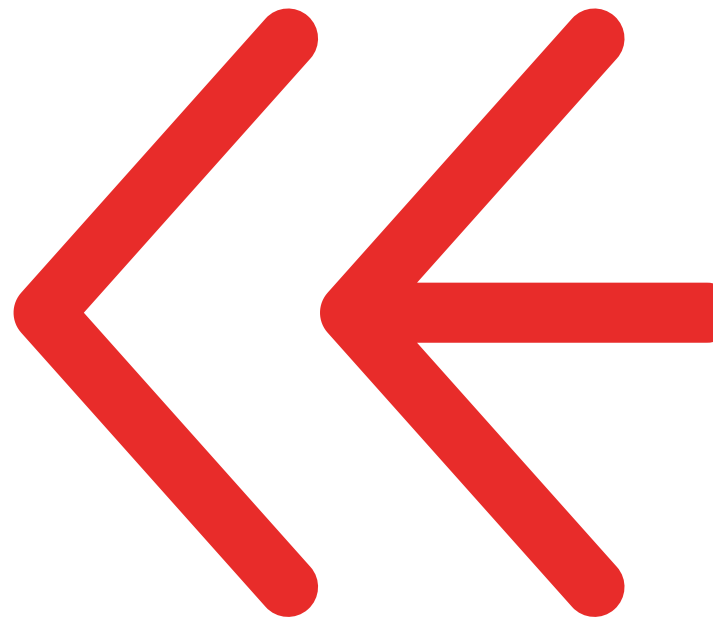
**A thriving DevOps culture means developers will trust their security counterparts to not slow down development cycles.**



Nothing is more frustrating to a developer than when security or legal is seen running interference by blocking further development until a security fix is found (which, in the worst case, can take days if not weeks to remedy).

“Developers often see security as slowing down development teams. While development teams within DevOps just want to get things done, they are not necessarily as worried about security as they should be,” Ivtsan said. “DevOps teams need to make sure that everything and all teams work together, and it is all automated. In fact, security is still not often being integrated or, if it is, it is seen as slowing down the development cycle.”





## DevOps culture still needs to shift more to the left.

All processes and participants must play an integrated part in DevOps from the very beginning—or in developer terms, at the left end of the production pipeline. Unfortunately, this is where the people part of the equation often represents a challenge for organizations that have established software development processes already in place.

As a remedy, it can make a lot of sense to convince developers to make the cultural shift if they see the vulnerabilities in the code at the beginning of the production pipeline, Ivtsan said. “The developers can also see, when security processes are integrated at the very beginning of the development process, that security processes will not slow development down.”

“Cultural changes should involve security teams integrating the security checks into the existing processes of development, including DevOps tools within GitHub and Jenkins,” Ivtsan continued, noting “much of the cultural friction that we see is also eliminated” when this happens.





## Automation should never create cultural silos.

There was a problem, yet the customer could not see the forest for the trees. Keyes remembers meeting a customer that had adopted the latest-and-greatest automation tools for DevOps and, the customer thought, a working DevOps culture. But there was just one catch: “When I first met this customer, they said: ‘We’ve been doing DevOps for 18 months, and have yet to ship anything,’” Keyes said. “They’ve certainly made progress since then.”

Many DevOps teams still believe that automation, while essential, will also create the necessary working culture to take advantage of the process. “A thought many developers have is, ‘Well, I’ll just automate it. And then it sort of brings us all together,’ when in fact what they’ve done is only automate the pipeline,” he noted.

Instead, DevOps necessitates the right working culture to allow automation to do its magic. A working DevOps culture will ensure that everyone involved relies on automation so that ultimately, code is developed, tested, deployed and managed in a way that software features benefit users faster and with fewer security vulnerabilities.

“Maybe an organization has done a great job of automating part of the delivery pipeline, but there’s a whole bunch of other things that still have to happen before customer value can be realized,” Keyes said.

# Communication remains critical.

Communication is absolutely critical for any relationship to survive—and DevOps is no exception, since enforcing a culture of communication is essential for an organization to embrace DevOps. This concept, in fact, was introduced decades before the advent of DevOps.

In this way, Conway's law still holds true. Melvin Conway wrote in his seminal paper, **"How Do Committees Invent?"** more than 50 years ago that "organizations which design systems (in the broad sense used here) are constrained to produce designs which are copies of the communication structures of these organizations."

David McKay, developer relations manager at InfluxData, said for DevOps today, organizations need to adopt a culture of understanding that the way software must be built and shipped should reflect the communication structure within their organization.

In other words, Conway's Law still applies. "So, how do you get started at a larger organization?" McKay said. "You have smaller teams building smaller services and you give those smaller teams autonomy to change the way they build and ship their services."



## Know how to communicate with those who can make the investment.

The issue almost invariably comes up: Someone in DevOps finds a solution for a business problem. The problem is that this requires an investment in a new tool or process. Maybe it is a new Jenkins tool that will take much of the headache out of uploading applications to the cloud so customers no longer suffer through shipment delays. Or maybe a new cloud offering will enable DevOps to offer end users more services at a lower price but will require additional investment in time and, initially, money. Many DevOps stakeholders struggle with making the suggestion—or worse yet, the communications structure is such that the right people never get the proposal. Again, this is where a culture of communication is critical.

Correcting DevOps culture in this case is often just a matter of knowing how to make the case to ensure the right tools adopted. “Developers often struggle with asking the correct questions in order to institute change with their supervisors. If you’re maintaining software that has none or limited tests, you may go to your supervisor and say, ‘Oh, we want to spend 20% of our time retrospectively adding tests for our software and we’re not going to add any new features,’ chances are you’ll get a resounding ‘No,’” McKay said. “However, if you frame it in an alternative way, focusing on the value first—‘We’d like to spend 20% of our time adding tests to our CI process so that we can deliver new features without breaking old functionality, with less bugs and much faster’—then that becomes a no-brainer. I don’t think I’ve ever met a developer, operator, manager or CTO that’s said ‘No, we don’t want to build and ship software faster.’”





## Rid the organization of 'design by committee' scourge.

As mentioned previously, software will be only as good as the communication structures of your organization, McKay said. The requisite cultural shift happens, especially for larger organizations, when smaller teams within the enterprise have the autonomy to make their own decisions. But what gets in the way are “the biggest time-sinks in our industry: meetings, exponential hand-off and design by committee,” McKay said.

“From the deployment aspect, what a lot of teams get right is continuous integration (CI), and then when they try to release, that’s when it all goes bad,” he added. “There’s still a lot of politics involved about who has the final authority: is it the product owners, or is it the developers or the operators? It’s always a political mess and it’s pretty consistent with every organization I’ve seen.”





**Your enterprise will never be the best or the worst at making the cultural shift to DevOps—but accept that improvements should be ongoing.**

After reading the above tips about how your organization can embrace a DevOps culture, chances are your enterprise has failed to adopt as least some of these practices. But rest assured that most organizations have not yet fully adopted a DevOps culture. According to the results of a [Gartner survey](#) published earlier this year, for example, 75% of DevOps initiatives will “fail to meet expectations due to issues around organizational learning and change.”




But, at the same time, it’s likely your organization has begun attempting to adopt a DevOps culture or plans to do so. “DevOps is never black or white—an organization’s DevOps is never completely mature, nor does it ever not work at all,” Ivtsan said. “It’s a process and it takes time to really do everything as you should.”

# Grow your Enterprise DevOps culture

Plutora connects data silos enterprise-wide to deliver complete workflow transparency. Promote trust and shared accountability essential to DevOps with visibility of team goals, responsibilities and their connection to business priorities.





-  <http://www.devops.com>
-  <https://twitter.com/devopsdotcom>
-  <https://www.facebook.com/devopscom>