



PagerDuty

# Moving to DevOps: The Human Approach

Adopting a DevOps approach can be daunting, especially for a skeptical team. Applying these five key pillars can make them converts.

# Executive Summary

The promises of faster innovation and deployment, better stability, and improved collaboration are motivating teams and organizations to take a closer look at how they can improve their existing operations by leveraging the principles of DevOps.

However, while the benefits are many, moving to a DevOps culture can be highly disruptive to the way teams work due to accelerated deployment velocity and the shift to being on call for responding to issues.

Many tout the benefits of DevOps, but the cold truth is that it's hard to do, especially for companies that have well-established processes, procedures, and tools. Then there's the elephant in the room: How do you manage this kind of change with your people without majorly disrupting your business? **How do you promote a culture of shared responsibility across the board and make DevOps a reality?**

Being on call is just one small part of DevOps, but it's often also one of the biggest hurdles to embracing a DevOps approach.

But don't lose hope: If addressed early on in your journey, it will actually help your team become the biggest advocates for DevOps and tout the benefits of on call.

Every organization is different, but in our experience, companies that successfully embarked on this journey followed five key tactics:

ONE

## Acknowledge Inevitable Change

TWO

## Create Buy-In for the Vision

THREE

## Understand the Change in Incentives

FOUR

## Make DevOps Your Own

FIVE

## Master the Metrics

ONE

# Acknowledge Inevitable Change

Every organization must innovate and change or risk becoming obsolete and irrelevant. But change is often uncomfortable, and for some people, moving to “you build it, you own it” might be one step too far.

And that’s okay. Some people may not be on board, and you may see some attrition as a result. It should be expected, and we’ve seen it happen in many organizations—after all, not everyone likes the idea of being on call! However, it might actually be better for your organization in the long run.

**You want people who are willing to help you get to where you want to go; those who don’t can slow you down. In this scenario, the short-term pain is worth the long-term gain.**

For those willing to go on the journey with you, help them visualize what it might look like.

Rather than making a wholesale change to a new way of doing things, start with a smaller project where people can test new ideas and new methods with less risk. The goal is to shift thinking so that on call becomes an opportunity to learn and grow instead of a barrier to change.

Provide training and experience on incident response and what it’s like to be on call, so they can see how it contributes to learning, growth, and quality. Plant the seeds for success and get some early small wins so that even if change is inevitable, it’s at least more familiar.

DevOps is not a zero-sum game; it’s additive. The intent is to continuously improve both the quality of the output, as well as the skills of the team.

TWO

## Create Buy-In for the Vision

Note that the shift to developer teams having more ownership can't happen without buy-in from the top as well, so it's important to ensure both management and development teams have a mutual understanding of the future state and potential benefits.

Starting small and getting some wins under your belt serve two purposes: First, it demonstrates that an Agile approach is achievable and works well for developer and ops teams alike. Second, getting this groundswell support makes it an

easier sell at the executive level because you can showcase these early successes.

Whether these successes are described by faster deployment frequencies and improved code quality on the dev side or fewer impact incidents and more resilient infrastructure on the ops side, it's crucial that these successes are material and visible for executive buy-in. From the dev and ops perspective, edicts from above (e.g., "go Agile now") don't do well, which is another reason that groundswell support is so critical.

Sparking this buy-in from your development teams may seem like the biggest obstacle to overcome, but their eventual support will become your most valuable asset. Developers will understand the value this transformation would bring to their role, and the executive team will see the positive business outcomes that result.

Committing to the vision aligns roles and responsibilities, and helps create a holistic approach to DevOps both from the top down and bottom up.

## THREE

# Understand the Change in Incentives

We hear a lot from our customers that using PagerDuty helps shift the development culture and enable more accountability. So what does this mean, exactly? In a traditional Ops model, incentives for developer and ops teams are typically misaligned. Developers want to ship fast, but have less visibility into reliability once code is in production. On the other hand, ops teams want reliability and perfectly running code, even if it means shipping slower.

A DevOps approach changes the incentives. Since developers own and fix the code in this model, they are much more motivated to improve quality even as they ship faster. There's no better motivation for creating better code than not being awakened in the middle of the night to fix it! That being said, a lot of developers fear being on call for this exact reason. But what we've found is that it happens a lot less than expected because the code quality greatly improves when using a DevOps approach.

**Being on call is the tactic that drives ownership home and aligns incentives. It also facilitates real-time learning, which drives better quality and faster innovation.**

FOUR

# Make DevOps Your Own

There's a lot out there about the principles of DevOps, the tools to use, and the practices to follow. But at the end of the day, the way you implement DevOps will be unique to your team and/or organization—and tools or processes on their own won't get you there. DevOps principles provide a framework; however, a successful transition requires a cultural shift—you can't just throw around DevOps buzzwords and expect the team to play along.

Actively consider how the team operates now, and how DevOps principles, tools, and processes can be tailored for how the team works. (If you'd like to use ours as a reference, we've made our [Incident Response](#) and [Operational Reviews](#) documentation publicly available).

We recommend taking a closer look at incident management in particular since it's likely that some developers will have concerns around being on call.

**Get input and incorporate team feedback, and make sure you truly listen to concerns and address them as you navigate how DevOps can be applied.** Test your new approach on a small project, adapt and incorporate key learnings into the next, and continue to iterate and refine.

Along the way, you'll gain more support and more adoption as you achieve more success, and the cultural shift will happen organically. It's a lot of upfront work, but the investment early on can pay dividends down the road.

FIVE

# Master the Metrics

To make a convincing argument about the benefits of DevOps, you need proof. Make sure you measure and quantify the existing processes to create a baseline for comparison by asking questions like:

- How long does it take to deploy a feature now?
- How long does it take to troubleshoot?
- How much downtime do you have?

You may also have additional KPIs with this approach. Our customers typically look at mean time to acknowledge (MTTA) and mean time to resolve (MTTR) issues for teams on call, but those are by no means the only metrics.

For example, many organizations measure team or on-call responder health as a way to understand and improve work-life balance, minimizing employee burnout.

## Sample KPIs to Measure

Metric	Why Track It?
Number of Deployments Per Month	Enables you to gauge developer productivity and CI/CD pipeline health
Service Availability	Provides insight into whether services or applications are accessible when required
Time Without Major Incident	Measures stability of core infrastructure
Alert-to-Incident Ratio	Shows how effectively related alerts are grouped to reduce alert fatigue and improve context and efficiency
Number of Major Incidents	Indicates frequency of business and customer impact
MTTA (Mean Time to Acknowledge)	Measures team response orchestration effectiveness
MTTR (Mean Time to Resolve) Major Incidents	Measures how quickly teams can troubleshoot and resolve major issues
Number of Interrupt Notifications	Provides early indicators for developer burnout and work-life balance by measuring notifications during sleeping, weekend, and off-work hours
% Regrettable Employee Attrition	Helps people managers understand the number of employees who leave due to burnout or preventable causes

# Conclusion

DevOps can have a wide interpretation, and what it means to your organization can be completely different for another. Moving to DevOps is a significant change that involves risk, patience, and commitment, and it can be jarring if it's done too fast or without buy-in. But with a thoughtful approach, you can alleviate many of the concerns and growing pains that can come up as you transition into a DevOps world with developers on call.

## ABOUT PAGERDUTY

PagerDuty is the leading digital operations management platform for organizations. Over 10,000 enterprises and small to mid-size organizations globally trust PagerDuty to improve digital operations, drive revenue, mitigate threats, protect assets, and delight customers.

To learn more about PagerDuty and sign up for a free trial, visit [www.pagerduty.com](http://www.pagerduty.com).



PagerDuty