

Beginner's Guide to Scaling DevOps





Over the years “DevOps” has become quite the buzzword. It’s been used to describe culture changes, automation, change management, continuous delivery, and essentially a culture where dev and ops collaborate to build a faster, more reliable release pipeline.

That’s all well and good... until you’ve been assigned the task of scaling your organization’s “DevOps transformation”. It’s an uphill battle of cultural resistance, disparate tooling, deep-rooted silos, and thousands of recommendations from across the web. Cue that sinking feeling in your stomach.

But fear not. Over the years we’ve learned a lot about scaling DevOps practices across an entire organization – not just from our own journey but also from our enterprise customers. Save yourself some struggle and start with these seven tips for scaling DevOps in the enterprise.

1. Don't force it

No one likes to be told what to do, especially if they don't know why they're doing it. Instead, show them how re-thinking the way they work can make them more successful.

At Atlassian, we use techniques from our [Team Playbook](#) to get teams on the same page and build a shared understanding of work to be done. Many of the plays are designed to improve collaboration, but two that specifically stand out to help with DevOps are [rules of engagement](#) and [retrospectives](#).

One word of warning, try to be mindful that not everyone in the organization will be interested in learning, which brings us to tip 2.





2. Do your homework

Arm yourself with specific examples from across the org where teams (maybe yours!) are working more closely together and enjoying the benefits. It doesn't have to be splashy. Even a modest reduction in rolled-back releases or customer-reported bugs is compelling.

Once you've found some success stories share them loudly and proudly with the rest of the organization. This could be through a blog on your company intranet, an email, or a lunch 'n' learn. Just make sure it's visible to as many folks as possible. And avoid chest-thumping. You want to inspire other teams – not make them roll their eyes.

At Atlassian we use features like @mentions and 'watching' to communicate better across teams. Not to mention a ton of internal blog posts on our Confluence instance. To learn more about how Confluence can support your DevOps initiatives, check out [The 6 best Confluence pages for building a DevOps culture](#).

3. Don't get hung up on the big picture

Have you ever heard the saying “The journey of 1000 miles begins with a single step”? Not only does it work for high school classroom inspirational posters, but it also describes the best approach to scaling DevOps. Too often, DevOps initiatives fail because they aimed straight for the moon without considering the individual steps to get there.

Start small. Identify the low-hanging fruit and replicate those changes first, for example, get all your dev and ops teams using a modern version control system like [Git](#), then move forward from there. And most importantly, don't forget to share your wins along the way.

When working with software teams at different levels of maturity, it helps to lay out the journey to DevOps ahead of time so teams can orient themselves. One of our customers, a Fortune 100 insurance company, rolled out a simple guide to help teams assess their practices, plan out a path to improvement, and decide which integrated set of supported tools would work for their teams.

4. Fake it till you make it

The best way to overcome skepticism is to get people doing the things they're skeptical about (maybe without them fully realizing they're doing it). Call it a hack or an experiment. Sometimes a little spin goes a long way.

A great example of this comes from a large telco who established a program for teams to practice applying DevOps techniques to their work in a guided series of Agile sprints to run a compressed timescale. The aim was for teams to experience what it feels like to work differently so that they could take lessons learned back and apply them to their workstreams.

5. Create consistency

In doing your homework, you may have noticed a phenomenon we hear about all the time from our larger customers: the same tools used independently by several teams throughout the organization. You may have uncovered 20 stand-alone instances of Jira Software, for example.

Consolidating into one shared instance is a great opportunity to provide some consistency. By doing so, you're building a common DevOps foundation where all teams can share best practices, easily access each other's work in progress, and create their own unique workflows without building more silos.

Customers use our tools to create a common collaboration layer where teams can plan, discuss, track, store, review, test, and deploy their work. And most of them integrate their Atlassian products with related tools like New Relic, Splunk, and Amazon Web Services.





6. Plan for rapid growth

Once folks start to realize the positive impact DevOps practices have on their daily routines, those practices will catch on. So plan ahead by choosing tools built to stay performant and stable against heavy usage and meet other requirements around security, compliance, and ease of administration and maintenance. Many of our largest customers choose our Server and [Data Center deployment options](#), which are built specifically to address these concerns.

7. When in doubt, get some help

Changing how teams work in large organizations is hard. Really hard. Getting help from an experienced DevOps practitioner can save you time, effort, and ensure you're getting the biggest bang for your buck. Fortunately, there's now a huge DevOps community out there, with loads of free resources as well as professional services offerings.

Our customers often begin with some internet-based research. Then, when they run into particularly thorny issues, they call in the pros: our 3rd party [Solution Partner community](#), which is highly knowledgeable in leveraging Atlassian tools and team guidance for DevOps.

DevOps is a journey, not a destination. There will always be ways to improve, new techniques to adopt, better collaboration techniques to implement, and so on. Getting your entire organization onboard with such an endeavor can be a whole other journey in and of itself.





Solution Partner

**Contact your
Solution Partner for
more info on DevOps**