



puppet

+



splunk&gt;

## 5 Stages of DevOps Evolution

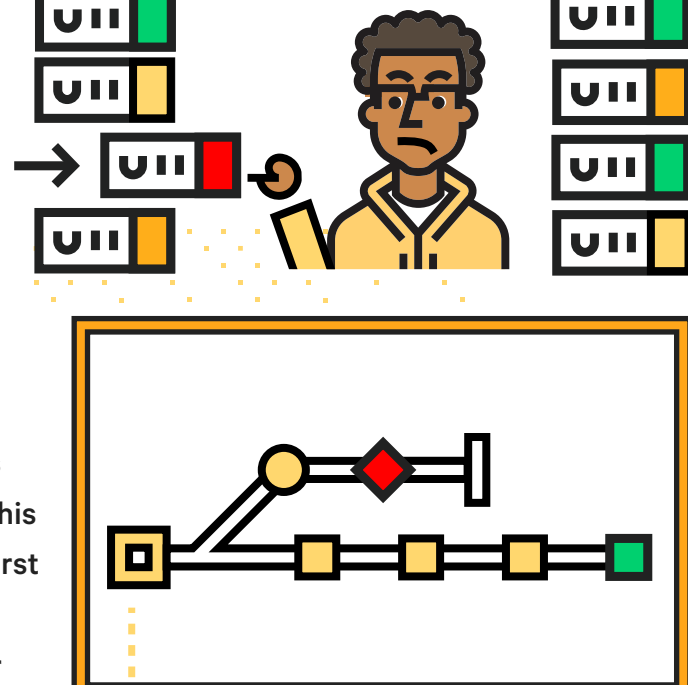
In our 2018 State of DevOps Report we surveyed over 3,000 technology professionals worldwide to give you a practical approach to adopting and scaling DevOps based on real data. While many organizations have achieved early success with DevOps, most haven't been able to broadly replicate and scale that success. Our data validates the anecdotal evidence we've seen over the years: there are defined stages in a DevOps evolution and specific practices can provide a shortcut to successful DevOps adoption. We also found that evolution is not linear — everyone has different starting points which can lead them down different paths — but there are prescriptive next steps to help you get on track and scale your success no matter where you are in the journey.

- 01 Normalize the technology stack
- 02 Standardize and reduce variability
- 03 Expand DevOps practices
- 04 Automate infrastructure delivery
- 05 Provide self-service capabilities

01

### Normalize the technology stack

At this stage, you may see the dev teams making a coordinated move to more agile development methods (e.g., an enterprise-wide Agile mandate), or a few teams organically adopting new methods for specific products or workflows. Dev teams at this stage have adopted version control, which is the first step on the path to continuous integration and continuous delivery. They're also normalizing their tech stacks by building on a standard set of technologies and deploying on a standard set of operating systems.



02

### Standardize and reduce variability

This stage is where teams concentrate on reducing variance and continue to standardize the tech stack by further reducing the number of operating systems (perhaps down to a single OS or OS family) and standardizing on a set of technologies.

This normalization and standardization reduces the overall complexity of the system, enabling teams to scale their expertise and apply consistent management and deployment patterns across multiple applications.

03

### Expand DevOps practices

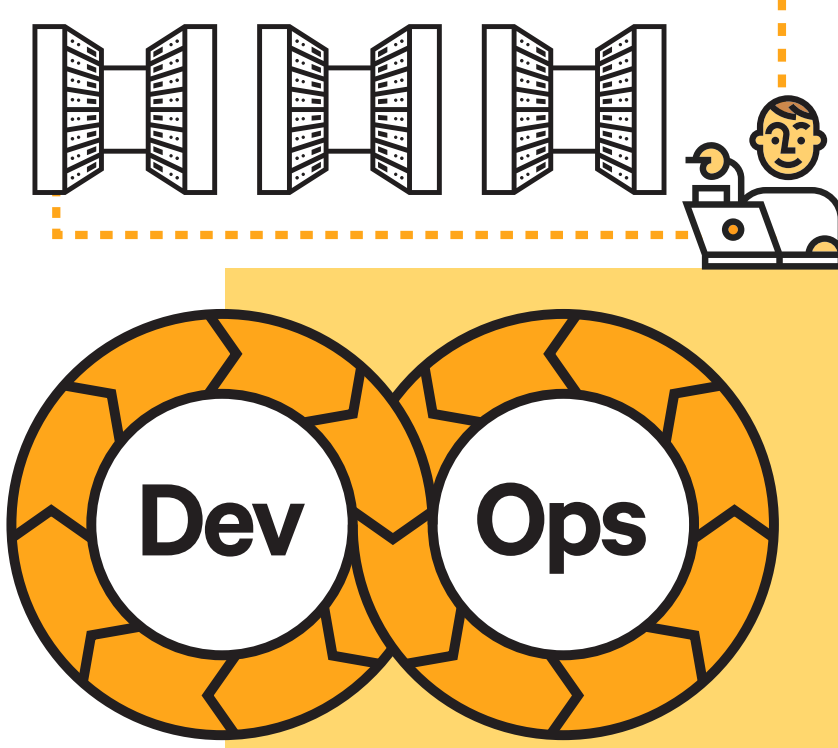
Now that the important foundational elements are in place, and the system is well understood, organizations can address other pain points. Typically, deployments are a huge source of pain and garner a lot of attention from management when releases are delivered late, or a critical defect makes it to production. Changes implemented in previous stages have caused application development teams' throughput to outpace the delivery team's ability to deploy. This discrepancy must be addressed quickly, or all the hard work at earlier stages will look like it made business outcomes worse, not better. Successful teams at this stage reuse deployment patterns for building applications and services, and infrastructure changes are tested before deploying to production. Both these practices provide predictability and reliability, building trust in the new methods and practices.



04

### Automate infrastructure delivery

This stage in the DevOps journey is defined by the automation of systems configuration and provisioning, and it's what many of us think of as the beginning of a DevOps initiative. Automating infrastructure delivery resolves the issue of developer throughput outpacing operations, and therefore the ability to deploy. Infrastructure automation certainly addresses a local pain point for IT operations teams, but it goes much further than that: it catalyzes the creation of self-service more broadly throughout the organization in subsequent stages.



05

### Provide self-service capabilities

By the time an organization gets to Stage 5, you can see the cumulative effects of achieving high levels of automation and trust. At this stage, resources are available via self-service, and incident response is automated. IT teams don't automate just for the sake of automating; they do it to make the entire organization run with greater efficiency and precision.

With self-service, teams across the business can work at their own pace, freed from manual approvals, handoffs, tickets and long wait times.

