
WHERE THE WORLD MEETS DEVOPS

[Home](#) » [Blogs](#) » [Leadership Suite](#) » 5 Biggest Ways to Fail at DevOps



5 Biggest Ways to Fail at DevOps



BY [CONTRIBUTOR](#) ON DECEMBER 19, 2017 — [4 COMMENTS](#)

The mission of DevOps is simple: Ship software, do it frequently, do it well and reliably. Repeat. Repeat.

Recent Posts By contributor

 [How to Ensure DevOps Success in a Distributed Network Environment](#)

 [Dissecting the Role of QA](#)

 [Show more](#)

Related Posts

 [5 Tips for Building a Winning DevOps Culture](#)

 [DevOps Connect: DevSecOps Edition Complete Session Videos](#)

Related Categories

 [Blogs](#)

 [Leadership Suite](#)

Every day, more organizations are turning to DevOps to deliver more frequent software releases, with the result that agile software delivery is gaining traction quicker than expected. Just how rapidly it is

Failing to Plan, Planning to Fail

But despite its growing popularity, the overarching concept of DevOps is still new to many organizations and things don't always go according to plan. Implementations can—and do—go wrong.

Here are the five biggest reasons why DevOps initiatives go south, and what you can do to fix them.

Failure No. 1: Not defining what DevOps means to your organization

The definition of “DevOps” is not always clear, and can have different meanings from one organization to the next. Developers may equate it with a specific approach to software builds, using popular tools such as Chef, Puppet, Jenkins, Git or Docker. Meanwhile, IT management might see it as a continuation of existing processes with an emphasis on faster time-to-market and lightweight release procedures.

For organizations launching rockets or running economies, “DevOps” means something very different than what it means to a startup. If a software release still involves the use of a change management tool such as BMC's Remedy, is it really DevOps? If it takes an hour to deploy a QA build, is it really DevOps?

Without a common definition, teams will argue over what DevOps is and why it's being used. Before introducing technology and processes under a DevOps initiative, organizations need to agree on a definition, be clear about what they are trying to accomplish and draw boundaries between DevOps and more structured approaches to IT service management.

Failure No. 2: Focusing on tools and techniques, while forgetting the teams

The second mistake many enterprises make is to elevate technology as the primary driver of DevOps at the expense of important processes that ensure quality software is meeting customer expectations.

It isn't enough to hire a few release engineers, provide them with virtual machines (VMs) and permission to install Jenkins and Puppet. For any initiative, the technology should not be the only consideration, people and processes also need to be taken into account.

More environments can always be created, but don't expect teams to scale overnight. Many companies adopt DevOps, moving to faster, more frequent releases driven by the needs of individual projects, before realizing that an increase in the frequency of software delivery can lead to QA and release management burnout.

If organizations are introducing more automation to accelerate their time to market, the impact of DevOps initiatives on the teams needs to be considered closely.

A lot of the rhetoric in favor of DevOps is rooted in the idea that developers need to take a proactive approach to evade change-averse administrators.

DevOps in the enterprise often emerges from one or two groups deciding to stage a “revolution” against an ineffective IT organization. When a company has a large system and intractable releases that take months, teams often lose patience with the process and are tempted to break the mold and move quickly.

The first teams to embrace DevOps were breaking the rules by design. They were comprised of independent teams free to innovate outside centralized IT structures. But while this approach works in smaller startups, it is not effective in many enterprises.

An organization with dozens of independent teams creating continuous deployment pipelines may sound good in theory, but it doesn't work so well in practice. DevOps isn't about reducing the number of governance gates. On the contrary, it enables more effective, more frequent governance gates to shine a light on complex release orchestration challenges.

Failure No. 4: Failing to account for risk

Many companies dive head first into DevOps without a full understanding of how it will affect risk associated with software releases.

When a company transitions from a slower, ITIL-focused process to a more agile, DevOps-focused reality, release managers are often expected to “wing it” toward the end of the release cycle. Many organizations forget to factor in risk for decisions about production. Software can be delivered faster, but the enterprise still requires governance gates.

Changes to production facing systems require rigorous change management, along with release management function to track conflicts and risk. One way to streamline the process is through continuous delivery management, which allows teams to conduct multiple simultaneous releases while integrating existing tools, that operations expect to have in place, to track and manage risk.

Failure No. 5: Running DevOps without metrics

DevOps teams often start out eagerly working to make change to an organization's infrastructure and release process, but they can also bite off more than they can chew.

Even though DevOps teams may want to reinvent a release process overnight, IT managers should still define metrics to evaluate whether initiatives are a success. Managing the slow transition to DevOps tools and techniques over several quarters can be difficult while prioritizing ongoing software development and release management.

and responsibilities to bring greater efficiency. It is essential to keep track of release and environment metrics to make informed decisions about particular initiatives from the DevOps teams.

As DevOps continues to grow in adoption and popularity, more organizations will seek to use it to speed software delivery and success. By avoiding these five common mistakes, they can ensure their initiatives have a better chance of success.

About the Author / Dalibor Siroky



Dalibor Siroky is the CEO and co-founder of **Plutora** with 15 years of leadership, consulting, enterprise product and operations experience across Australia, Asia and Europe. Prior to Plutora, Siroky was founder and managing director of Finotaur, provider of independent management consulting services, and CIO of financial advisory software at Macquarie Bank, head of solution architecture at Commonwealth Bank of Australia and management consultant at PricewaterhouseCoopers. Siroky received his MBA from the University of Chicago Booth School of Business. Connect with him on [LinkedIn](#) and follow him on [Twitter](#).

Sponsored Content

Featured eBook



The Seven All-too-common Barriers to Successful DevOps Transformation

While many companies attempt DevOps transformation, not all are successful. Learn how enterprises have overcome the most common barriers to success. Today, every enterprise is critically dependent on the software and online services that they run. Those that learn to excel at software delivery will excel at their business. The guidance ... [Read More](#)

FILED UNDER: [BLOGS](#), [LEADERSHIP SUITE](#)

TAGGED WITH: [DEVOPS FAILURES](#), [DEVOPS INITIATIVES](#), [IMPLEMENTING DEVOPS](#)



106



More

[« Persistent Storage: You'll be Hearing More](#)[Real World Development »](#)

4 Comments DevOps.com

[Login](#) ▼[Recommend](#) [Share](#)

Sort by Best ▼



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

**Steven Morris** • 17 days ago

A home run of an article, kudos.

^ | ▼ • Reply • Share ›

**Keith Hoene** • 4 months ago

Defining what it is and isn't is the biggest at my work. We have the 'rush to automate' and thoughts that we can take SMEs and put them in an automation bootcamp for a week and come out Engineers without first examining if a team even has a framework to build on. Love the over emphasis on tooling as I am seeing everywhere DEVops being used as a huge opportunity for a tool grab.

^ | ▼ • Reply • Share ›

**IQ Online Training** • 4 months ago

Hey Dalibor,

I love your work,so much so i even linked out to you in my latest Quora blog post.

Feel free to check it out :- <https://www.quora.com/What-...>

Cheers,

Rahul

P.S if you share this post,it wouldn't just make my day

But a great year.

Thank You!

^ | ▼ • Reply • Share ›

**Helen Beal** • 5 months ago

Great article! I'd also add not engaging the business or understanding or articulating the value of DevOps improvements.

^ | ▼ • Reply • Share ›

[Subscribe](#) [Add Disqus to your site](#) [Add Disqus](#) [Add](#) [Privacy](#)

NEWSLETTER SIGN-UP

Notice: Submission of this form includes an automatic

subscription to the DevOps e-newsletter. [DevOps.com Privacy Policy](#)

SUBSCRIBE

UPCOMING WEBINARS

THU
10
Winning at Culture: The Keys to the Successful DevOps Organization
Thu, May 10, 1:00 pm - 2:00 pm EDT


TUE
15
Cloud Migration: Solving for Growth with Containers & DevOps on AWS
Tue, May 15, 1:00 pm - 2:00 pm EDT

WED
16
Bridging The Gap Between Dev and Ops
Wed, May 16, 1:00 pm - 2:00 pm EDT

[Past Webinars](#) [More Webinars](#)

DOWNLOAD FREE EBOOK

Download

 **DEVOPS CHAT**
DOES London 2018 Preview w/ Mirco Herring, DevOps for the Modern Enterprise

[Compuware](#)

[Low-Code Is Coming To An App Near You, Appian World Preview](#)

[Microservices on Google Cloud Update with Tim Hockin](#)

[DevOps Chat: Interop China Features](#)

[DevOps & DevSecOps](#)

PAST WEBINARS



[HOME](#) [BUSINESS DIRECTORY](#) [ABOUT DEVOPS.COM](#) [MEDIA KIT](#) [SPONSOR INFO](#) [COPYRIGHT](#) [TOS](#) [PRIVACY POLICY](#)

© 2018 · Mediaops, LLC.

DevOps.com

- [Home](#)
- [Business Directory](#)

- [Sponsor Info](#)
- [Copyright](#)
- [TOS](#)
- [Privacy Policy](#)

•

Search DevOps...

•

•

•

○

○

○

○

○

○

○

○

•

○

○

•

•

•

•

•

•

•

○

○

○

○

○

○