



r/ devops



r/ devops



Search Reddit

Get App

Posts

r/ devops



259



Posted by u/MrScotchyScotch 9 months ago



DevOps is dead, long live DevOps

People don't get the term. The entire tech industry's misunderstanding of the term is, quite frankly, embarrassing, tiresome, and getting in the way of progress. I think it's time for it to die.

We know DevOps is not a role. But of course there are thousands of roles with "DevOps" in the title. We know DevOps is not one thing you "do". But people are constantly asking - or telling - how they "Do DevOps" or "Get Into DevOps" or "Become A DevOps".

Each role doesn't seem to understand what the term means. Executives to low-level Engineers, Security to Operations to Development to QA to UX to Data Science. They all think the word means somebody who writes Terraform and gate-keeps the AWS IAM Administrator role. Common use of the term implies it means "setting up server software in Linux". And most of the roles listing "DevOps" also imply just Systems

About Community



r/ devops

Welcome to devops



Created Aug 2, 2010

254k

Members

402

Online

Join

Similar to this post



r/coolguides



Devops recruitment post in LinkedIn shows what devops...

65% 0 · 10/22/2022



r/ devops

DevOps Is an Evolving Culture, Not a Team

89% 54 · 3/29/2019



r/ devops

Administration skills with cloud-based technology. Add in a few buzzwords like IaC and Immutable Infrastructure, and that's all there is to it.

It is so completely misunderstood by 99.99% of people that almost nobody uses it in the proper context. The only people that do are the *tiny, tiny few* that have actually read all the books and blog posts and gone to the conferences. Most people will never understand DevOps. Which would be fine, if the people who are *hired to do DevOps* actually understood how it worked.


Of course, a select few know DevOps' real definition ("***a combination of specific practices, culture change, and tools intended to shorten SDLC by reducing the time between committing a change to a system and it being continuously delivered into normal production while ensuring high quality and reliability***"). But like Agile, Lean, Six Sigma, etc, the definition alone doesn't tell how it works. It only leads towards a series of rabbit holes needed to learn the many different concepts, still without revealing how to implement them.

Unless you are a consultant working on Digital Transformation, you won't learn what DevOps actually encompasses, and probably will never work on all aspects of it as an IC.

This perpetually-misunderstood nonsense word will continue to be a blight on the practices it is intended to push. I think we need to take

DevOps Role.. but not really DevOps?

↑ 33% ↓ 10 · 11/16/2022

 r/azuredevops

DevOps Pipeline - Confusion in variables structure

↑ 100% ↓ 6 · 9/29/2022

 r/nier



Devola and Popola from NieR Automata by @m1ne0h

↑ 99% ↓ 56 · 5/11/2021

 r/devops

CircleCI Security Incident

↑ 95% ↓ 35 · 2d

 r/devops

Developer decided to change project IaC tool while I was...

↑ 92% ↓ 112 · 16h

 r/devops

Terraform - How do you handle secrets?

↑ 98% ↓ 68 · 3d

 r/devops

Monitoring infra cost: which tool do you use?

↑ 89% ↓ 49 · 4d

 r/devops

Linux System Config Management Tools

↑ 96% ↓ 81 · 6d

action now to stave off the industry from continuing to fail at implementing it.

- We need to kill the term, so that new terms that speak to more specific aspects of DevOps can arise. Then it will be easier for people to be aware of them, learn how they work, and try to implement them.
- We need to remove the DevOps role and replace it with multiple bodies of knowledge aimed at different existing tech roles. DevOps should be implemented by many different roles at the same time, in unison. We also need to avoid roles that merely gatekeep access to production services/accounts, and focus on building the platforms that enable multiple roles to achieve system-level functionality without becoming systems experts.
- We need to remove the cargo-cult aspects of DevOps buzzwords and develop real engineering disciplines based on DevOps practices.
- We need to show teams real world examples of DevOps culture that achieve both trust and excellence in the production and operation of software systems.
- We need a manifesto (akin to the *12 Factor App*) with a set of rules for how to adopt DevOps into a team and design a system

 r/devops

How do you manage configuration/secrets changes?

↑ 93% ↓ 25 · 5d

 r/devops

ClickOps?

↑ 82% ↓ 103 · 1d

 r/devops

How to scale artefactory for 2000-3000 users? Possible?

↑ 90% ↓ 37 · 5d

 r/devops

Scaling a web application which is growing very fast

↑ 91% ↓ 18 · 1d

 r/devops

Moving from openlens to k9s after the 6.3.0 downgrade

↑ 88% ↓ 12 · 1d

[Top Posts April 11th 2022](#) [Top Posts Of April, 2022](#)
[Top Posts 2022](#)

[User Agreement](#)
[Privacy Policy](#)

[Content Policy](#)
[Moderator Code Of Conduct](#)

Reddit Inc © 2023. All rights reserved

[Back to Top](#)

with it in mind.

- We've gotta stop calling it DevOps first, though... so we need new ways to refer to those specific components of DevOps without using the word "DevOps".

I'm happy to propose some of these changes myself, but I'm hoping others have already started down this road and can provide some guidance.

80% Upvoted



184
Comments



Share



Save

What are your thoughts?

B

i

o

S

comment

A



T

≡

≡

99

...

Sort By: Best ▼

Octal · 9 mo. ago 

Hey where I am companies pay through the teeth for "DevOps" personnel, so please stop trying to destroy the gravy train!



338



Reply

Share

Report

Save

Follow

Merejo · 9 mo. ago

my first thought, I am
getting paid well to be a
"DevOps engineer" so stop
this nonsense LOL

 **45**   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

[Continue this thread →](#)

gutigen · 9 mo. ago

This.

As long as I get that
"DevOps" salary, I could be
called Junior Janitor for all I
care.

 **99**   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

[Continue this thread →](#)

drakk0n · 9 mo. ago

I'm envisioning a Spartacus
moment here - "I am a
DevOps Engineer"

 **29**   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

[Continue this thread →](#)

hatchikyu · 9 mo. ago ·
edited 9 mo. ago

Be careful what you wish for.

I joined a non-tech field in the mid-2000s that was hot like what tech and DevOps in particular is right now. It had the same markers DevOps has: well-respected, high-growth market, ambiguous work patterns, perceived mission-criticality, scarce talent pool etc.

Powers that be will always try to find ways to get workers for less cost i.e. moving the gravy train in their direction. And it will happen at some point.

That non-tech professional field I mentioned earlier is now a husk of what it was in the late 2000s. Weak pay relative to less skilled jobs and more work without any of the integrity and culture refinements that OP is pushing for.

Because nobody bothered.
Because the money was good. At the time.

 8   Reply  Share  Report  Save  Follow

Neil_Fallons_Ghost · 9 mo. ago

Honest question as someone in lead devops role. What's the pay packages youre seeing?

 4   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

techworkreddit3 · 9 mo. ago

Yepp, I don't care what you need to call it, just keep paying me this rate.

 13   Reply  Share  Report  Save  Follow

SexyMonad · 9 mo. ago 

They all think the word means somebody who writes Terraform and gate-keeps the AWS IAM Administrator role.

And I took that personally.

 331   Reply  Share  Report  Save  Follow

royalme · 9 mo. ago

Consider, if only for a fleeting moment, if you are the one who doesn't get something.

Devops is only a term, an idea, that someone created. It's not a mandate from heaven. People (or companies) can pick parts of the idea they like and want to follow. And they're not wrong for only following parts of it if it fits their goals.

In fact it's an idea that's well known to partly overlap with concerns of an SRE. Both deal with concerns of managing technology operations.

There's lots of ideas in the industry. There's Agile, there's Scrum, there's OOP, and functional. There's TDD, pair programming, domain driven development. We can't even agree on the definition of a unit test vs an integration test.

Practically nobody is doing any of it correctly, and nobody is doing all of it. They are not mandates, they are just ideas. Don't get hung up on one and live in the woods and write a manifesto or anything.

 **147**   Reply  Share  Report  Save  Follow

heavyrain123 · 9 mo. ago

Solutions Architect / Cloud & D...

I agree with you, but I think what OP is saying is that most of the industry is doing it consistently and systematically wrong. It's not just that they're doing DevOps wrong, but they're doing it wrong the way everyone else is doing it wrong and not asking the right questions.

Doing DevOps the wrong way has become industry standard, which means of course the reality of devops is divorced from the theory of devops, meaning all of our jobs are that much harder.

I'm constantly facing this problem where customers are implementing CodeDeploy/IaC to "do devops" and then wonder why 7 people from different silos need show up to a meeting to decide how the infrastructure should work which inevitably causes delays and friction. It's hard to explain the culture/Conway's Law angle of devops to them because they've never heard about it before from anyone else.

 26   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

[Continue this thread →](#)

qub3r · 9 mo. ago

A couple views I would add are: 1. DevOps will inherently look or be implemented differently across organizations. 2. DevOps describes a path/journey where the tools, processes, and culture will be more or less mature depending on where they are.

For example, a single person doesn't wake up one morning, decide they are now going to follow a minimalist lifestyle, and have that be the start and end of it. It happens over a long period of time. This is even more so the case when you're talking about a larger group of people.

One can look at a static snapshot of an organization and be able to point out a lot of things that could be improved, but it's imperative to understand: what's the focus and how do the decisions being made in the areas of focus line up with the commitment to DevOps?

 **3**   Reply  Share  Report  Save  Follow

PartemConsilio · 9 mo. ago

I have been hired as a “DevOps engineer” and when I was hired I even explained to my current employer this isn’t really a “thing”. But you know what? I decided that I’m going to own it and I believe part of my role is consulting on design and methodology. So far, I have been received well for that. I often tell people my basic role is “I remove bottlenecks for development so devs can focus more on building shit and ops can focus more on automating shit.”

 **42**   Reply Share Report Save Follow

zhynn · 9 mo. ago

This is what DevOps is, and it is not an illusion.

I see it as automation between code and infrastructure. It can touch lots of different specializations, but it is really just implementation of automation between software engineer and infrastructure. That infra may be cloud, but it could also be on-prem.

Good DevOps ends up being a force multiplier for the other more specialized roles (software engineers, SREs, technical admins, customer service... even sales in some cases). And it's also a role that bridges worlds, so it works well in a consultancy role as well (as you mentioned).

We are "doing it wrong" in that we used the term to be a placeholder for those responsible for the infrastructure and CI/CD instead of the application code itself. But the management of the infra is mostly irrelevant, it's the automation tooling that we build that really adds value.

↑ 11 ↓  Reply  Share  Report  Save  Follow

rcls0053 · 9 mo. ago ·
edited 9 mo. ago

Nice rant, but not very new. The same thing has happened with many other concepts, like agile. Some people take up their torches and start spouting how we need to kill this and that to get a fresh start. It's not gonna work. It's the nature of the business. However, us people who understand how wrong they are can educate them. Even now I'm teaching DevOps principles and culture to a customer who already set up a team to experiment with, to use those practices and see it's outcomes. They've embraced continuous learning and are moving in the right direction.

 **79**   Reply Share Report Save Follow

OlandoSC · 9 mo. ago

It can be more impactful to lead by example! I am in a similar position where I am helping an organization shift toward what you may call a 'devops culture', enabling teams to deliver their product more effectively, creating meaningful feedback loops to quickly integrate changes etc.

This org isn't in the software development space and they aren't using a ton of tools one might call 'devops' tools, but they have begun to embrace the ideas that enable the team to affect change.

 **4**   Reply  Share  Report  Save  Follow

smcarre · 9 mo. ago

I think one key difference between this and agile is that agile is an actual working framework set up by a specific group of people that defined what is and what isn't agile, DevOps isn't.

 **2**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

[deleted] · 9 mo. ago

You are trying to change the world. Does nothappen. Right or wrong doesn't even matter You will.keep.wasting your energy on things you can't control. Best wishes

 **123**   Reply Share Report Save Follow

ThoseWereTheDays · 9 mo. ago

This +++

 **5**   Reply Share Report Save Follow

[Continue this thread →](#)

NormalUserThirty · 9 mo. ago

wise

 **1**   Reply Share Report Save Follow

rickerdoski · 9 mo. ago

...says the million lemmings that are supposed to never be wrong.

 **-2**   Reply Share Report Save Follow

luenix · 9 mo. ago

System Engineer

> new terms that speak to more specific aspects of DevOps can arise

Leaving out programming and infosec for a moment and just rattling off a few:

- SDET
- SRE
- DBA
- Systems Integrator
- Systems Engineer
- Product Manager
- Systems Analyst

 **35**   Reply  Share  Report  Save  Follow

Abhir-86 · 9 mo. ago

Release engineer

 **25**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

smcarre · 9 mo. ago

YAML developer

 **4**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

Robert_Arctor · 9 mo. ago

Platform engineer here - I like it better because it describes the role much more clearly. I work on the scaling, monitoring and upkeep of our cloud platforms.

 14   Reply  Share  Report  Save  Follow

ikkkkkkkky · 9 mo. ago

Build Engineer

 4   Reply  Share  Report  Save  Follow

melisargh · 9 mo. ago

Linux lover

"SRE" here I'm like a third level of incident escalation? Also automate some stuff, honestly I'm not happy here so I'm migrating to pre sales, at least if I'm a lie I get to make a profit about it

 0   Reply  Share  Report  Save  Follow

FreeYellow6768 · 9 mo. ago

destroyer

 1   Reply  Share  Report  Save  Follow

Dr_Pills · 9 mo. ago

Add cloud engineer

 1   Reply  Share  Report  Save  Follow

EiKall · 9 mo. ago

Bad rant, not enough shouting.
3/10.

Also Gartner told us that
DevOps is dead years ago. Now
you have to buy at least
BizDevNetSecOps to stay ahead
of the hype cycle. /s

At the end of the day good
workers want to work better
instead of harder, so they look
for relevant keywords in job
offers. So now every HR SEO
slaps a DevOps on their job
offer, no matter what they want
you to do.

 18   Reply Share Report Save Follow

t_go_rust_flutter · 9 mo. ago

The management in my old
company thinks DevOps means:
all software engineers know
everything about development,
testing, producing and
deploying software. Including
setting up and maintainig a
complete CI/CD pipeline.

Heck, Microsoft teaches that
this is what DevOps is for Azure.
The AZ-204 certification
assumes (or at least implies)
one person should do
everything from developing C#
to maintaining the CI/CD
pipeline.

 28   Reply Share Report Save Follow

lorarc · 9 mo. ago

YAML Engineer

What is wrong with devs maintaining the CI/CD pipeline? I don't write the tests for them, why would I care about how the tests are run?

CI/CD pipeline should only be concern to the ops/devops when it comes to maintaining the servers it runs on but we should aim for having it as PaaS.

 **29**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

No-Safety-4715 · 9 mo. ago

This *is* the core goal of DevOps: that everyone is both a full fledged developer and knows operations. There is no other way to reach the DevOps ideal without this concept. *BUT* that's simply not possible in today's world. There is too much overhead for any single individual to master all those skillsets *and* be able to maintain them. Some day, cloud and automation may become so refined it's achievable, but right now, it's simply too much work.

 **12**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

rcls0053 · 9 mo. ago

I'm a developer and I can do this, no problem. Even now I'm setting up automated tests in a customer's pipeline.

You don't need to know everything. Even I have no idea how to do networking in the cloud or off the top of my head set up a load balancer in front of multiple servers. There are three major cloud platforms, and also PaaS services, so it's kind of difficult to know "everything". But it's really recommended that developers start to get to know these things to and understand their value. It also helps to embed Ops people into teams to introduce developers to these things so they start learning how to use those tools.

 **18**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

coinclink · 9 mo. ago

Honestly, as long as you have a good template for new projects as they arise, the developers *should* manage the CI/CD pipeline. With CodePipeline, it's even possible to manage the code for the pipeline right in the same repo as the application code.

I mean, what is it other than some YAML and basic shell scripts defining a workflow? Should a C# dev not be able to do that??

 **2**   Reply  Share  Report  Save  Follow

GeorgeRNorfolk · 9 mo. ago

Software engineers should know *a bit* of everything, that's what being T-shaped means. We should all have a base knowledge set of how everything works, and then a specialism in one of the areas. A software engineer specialising in DevOps should know a bit about how to write javascript (assuming the app is JS based) and automated tests.

 **2**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

Obsidian743 · 9 mo. ago

Yes...and?

It's the next evolution of the "full-stack" engineer. That's why we are in high demand and get paid a lot of money.

 **0**   Reply Share Report Save Follow

chuchodavids · 9 mo. ago

DevOps is a word; in linguistics, a word evolves. Just like the word **bizarre** was misused so bad that it had to be redefined, the DevOps word is going to be redefined too. Words mean what the people that use them makes them to mean. You can fight back and complain, yes. But that ship is long gone baby.

 **21**   Reply Share Report Save Follow

Acelection · 9 mo. ago

Thank you, this is the comment I was looking for. Language is not a static thing and fighting this will only make you miserable

 **11**   Reply Share Report Save Follow

MrScotchyScotch **OP** · 9 mo. ago

I agree that we can't change the shifting meaning of a word. That's why I want people to stop using the word, and for us to find new ones to express what we actually mean. If we don't, then we will never get across the new ideas. We need specific words to understand ideas clearly.

 **1**   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

hugthemachines · 9 mo. ago

First you have to ask yourself:
"Do I prefer banging my head against a wall or do I prefer doing something constructive."

when you have worked in this business for a bunch of years you will notice you should pick your battles.

 **5**   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

forsgren123 · 9 mo. ago ·

edited 9 mo. ago

I think the largest issue is that most DevOps guys are just hands-on implementors (ICs) who are mainly interested in writing app/infra code, working with cloud platforms and Linux administration. Whereas teaching and implementing DevOps culture would require a senior consultant with good influencing and leadership skills (what many here would call hand waving) or alternatively you would need to be CTO at the company to have mandate to change things.

But I do see Platform Engineering coming up more and more, and I think that is eating away at the whole DevOps concept. Atleast then we shouldn't have these reoccurring rants anymore? :)

 **16**   Reply  Share  Report  Save  Follow

killz111 · 9 mo. ago

Platform engineers are second degree gate keepers for DevOps. Increasing I see the market of DevOps that don't want to know networking, change management, WAF, etc. So now we got opinionated stacks write yaml for you base on slightly leaner yaml.

It's just shifting the problem to a different group.

 **2**   [Reply](#) [Share](#) [Report](#) [Save](#) [Follow](#)

thinkmassive · 9 mo. ago ·
edited 9 mo. ago

Check out the book "Team Topologies"

It describes exactly what's encompassed by DevOps without ever needing to say it outright. It also provides high level descriptions of "platform" and SRE, in a way that makes them much easier to discuss.

Convince others in your org to read the book and you'll establish a common vocabulary, so everyone can concentrate on efficiently shipping software instead of throwing crap over walls and waiting for the next fire.

Follow-up: They have a website too, although it has "bonus" content like \$\$ video courses and stuff. I haven't tried it because it's expensive, and the book is plenty descriptive for teams that are already successfully practicing DevOps methodologies. If you're struggling in a larger org or have a lot of clueless coworkers I could see the value though.

<https://teamentopologies.com/>

 4   Reply  Share  Report  Save  Follow

[deleted] · 9 mo. ago

Well, if 99,99% of people do something the same way, it becomes the normality, so you are wrong now.

↑ 10 ↓  Reply Share Report Save Follow

moduspol · 9 mo. ago

This is *literally* the case.

↑ 1 ↓  Reply Share Report Save Follow

BuddahsThumb · 9 mo. ago

[\$firstTimeNooseMeme]

we need to kill the term

Unless you have the ear of several of the largest PR and marketing firms, not going to happen. Literally billions of dollars of perceived market value has been pegged to the DevOps term.

We know DevOps is not a role

Who are these “we” you speak of? Techies? Our opinion hasn’t counted in the grand scheme of things, since the wheel was discovered.

The entire tech industry’s misunderstanding of the term is [...] getting in the way of progress.

Explain what progress is being impeded?

This perpetually-

misunderstood nonsense word will continue to be a blight on the practices it is intended to push.

Companies have needs. The vast majority of them really don't give a hoot about what you call it, they want their shit running reasonably well, for the lowest possible cost, in the shortest possible time. It isn't a misunderstanding of the word "DevOps" that's the problem, the issue is a misalignment between what those who pay money for services want and need, and the craft *you* want to practice.

I'm happy to propose some of these changes myself,

Propose to whom, the Elders of the Internet? Is there some relevant governing cabal I didn't get the memo about?

Seriously, I appreciate the sentiment, but of all the problems you can pick to get excited about, misunderstandings about what "DevOps" exactly means is nkt very productive.

 26   Reply Share Report Save Follow

[deleted] · 9 mo. ago

What is the point of this post?
DevOps is developer operations,
we don't need some pretentious
monologue about how you're
changing culture or changing
the world or who knows what.

 8   Reply  Share  Report  Save  Follow

No-Safety-4715 · 9 mo. ago ·
edited 9 mo. ago

The core reality of your
complaint stems from the fact
no matter how nice the ideal of
true DevOps philosophy is, in
the real world, everyone doesn't
have time to perform all the
tasks that would be required of
them and the work keeps
getting split up.

A lot of DevOps is aimed at
removing the silos of devs and
ops and trying to, essentially,
give devs the keys to the
infrastructure and pipeline so
they can be more efficient. The
cold reality is that's not really
possible to fully do in most
situations. The old Ops roles like
sysadmins existed because
there is simply *so much*
overhead involved. Knowing the
tools, maintaining the systems,
monitoring, etc. it all takes time.

DevOps philosophy is in some
ways looking for unicorns. As
automation improves and cloud
has grown, its core goal seems
near achievable and maybe in

the near future it will be as more things integrate, but for now, it still takes specialized focus to roll out infrastructure. That means asking developers to be masters of so much software and skillsets that it's just not realistic for them to fully do both Dev and Ops at high level.

So, my point is, you have to come to terms with *why* DevOps as an ideal is not really being achieved by companies. The workloads to reach the ideal are still too high, but that might not be the case in another five years.

 3   Reply  Share  Report  Save  Follow

Obsidian743 · 9 mo. ago

A-fucking-men!

I've been preaching this over and over on this sub. Sadly, most of the people here are just ops people who think "dev ops" is them being embedded on a cross-functional team instead of a separate ops teams.

If you're an ops person who isn't in the application code or an app developer who isn't in the pipeline you're *likely* not dev ops.

 3   Reply  Share  Report  Save  Follow

Fodagus · 9 mo. ago

I do and don't agree. I have a title of "DevOps Engineer", and what I see my role as is to "engineer" the "DevOps" at my org. This means, yes, being the team bitch sometimes, but also thinking about how we can improve developer workflows, advocating for better practices, and writing the automation that powers it all. And yes, a huge helping of SysOps albeit abstracted into Kubernetes and Ansible. This is a full time job. Sure, if my devs were better at writing Dockerfiles and understood Kubernetes, there'd be less to do, but if just do more projects around improving security scanning, adding better runtime metrics, etc.

Beyond this, my team's long range vision includes making improvements to standard packages that our dev teams would use to cut down boilerplate, and keeping a mind more in the game on how these services will scale and what we need to enable that.

My experience has shown me that a shockingly few number of Devs care at all about the infrastructure. I don't mean the servers, I mean all of it. They want better development environments, automation, and durable deployments, but they don't really want to *do* it. Maybe that speaks to the quality of devs I've worked with some, but I definitely knew a few very

good architects who hate doing anything that's not writing their software.

tl;dr. As a DevOps Engineer I don't "do DevOps," I advocate for DevOps and drive the vision behind our take. Kinda like how a Release Train engineer didn't do releases, but sets the culture and improves the process.

 3   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

AccurateVictory · 9 mo. ago

a cloudy term for a cloudy world. :)

 3   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

shyneep · 9 mo. ago

I think platform engineer is a better catch all term for what we do.

under Software Engineer:

- product engineer -- mobile, web, backend, frontend etc.
- platform engineer -- cloud infra, sre, security, release, etc.

 3   Reply [Share](#) [Report](#) [Save](#) [Follow](#)

BeaconRadar · 9 mo. ago

The responses in this thread just confirm your point OP. This is a sub mostly frequented by ops(role) turned DevOps (Role) when their company moved to the cloud. We might need a new sub [r/DevOps2](#)

 3   Reply Share Report Save Follow

duebina · 9 mo. ago

DevOps today is synonymous with the old school notion of combining a system with a developer and then only offering enough pay for the lower paying title.

Except the market pays. As it should.

 6   Reply Share Report Save Follow

ptownb · 9 mo. ago

You sound salty

 8   Reply Share Report Save Follow

its_that_dude_ · 9 mo. ago

If everyone is an asshole, you're the asshole

 6   Reply Share Report Save Follow

killz111 · 9 mo. ago

LoL the funny thing is DevOps engineers can't change culture. Only management can.

If you go back and read the Phoenix project what was the single most important change that put the team back on track for success? It's none of the fancy concepts. It was the fact that they put an ops guy that was not ambitious, genuinely cares about delivering the right outcomes and willing to stand up to make things better.

A bunch of overpaid engineers aren't gonna change a company culture. Only changing management culture can do that. This is why Toyota succeeds but the Toyota way gets killed when they tried to bring it to GM.

So stop focussing on the term and start holding your management to account.

↑ 4 ↓  Reply Share Report Save Follow

jona187bx · 9 mo. ago

Yep. Change always has to come from the top or you are just skating up hill.

↑ 1 ↓  Reply Share Report Save Follow

attitudehigher · 9 mo. ago

Hey bro... do you even DevOps?

 **8**   Reply  Share  Report  Save  Follow

8ersgonna8 · 9 mo. ago

Most organizations and recruiters still are still too oblivious and label it as “devops” so I will continue using it as a title for maximum LinkedIn exposure. Might change it to SRE soon though.

 **3**   Reply  Share  Report  Save  Follow

zpallin · 9 mo. ago

The whole point of the title SRE was to create a specific methodology behind that role versus the DevOps engineer whose purpose is entirely nebulous. I've had to argue endlessly with upper management to make it clear that you cannot hire an SRE. You have to create an SRE team and then hire people for that, otherwise SRE just becomes a synonym for DevOps Engineer.

 **1**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

daedalus_structure · 9 mo. ago

It is so completely misunderstood by 99.99% of people that almost nobody uses it in the proper context.

The concept is so ill defined that nobody understands it.

Also, as a general rule, when someone tells you they are one of the chosen few who understand something it is always more likely that they don't understand it either.

We need to remove the DevOps role and replace it with multiple bodies of knowledge aimed at different existing tech roles.

Why do none of these long winded manifestos ever address the realities of managing cognitive load for development teams that are already stretched thin?

We need a manifesto

Please god no more manifestos about the one true way.

 4   Reply  Share  Report  Save  Follow

hatchikyu · 9 mo. ago ·
edited 9 mo. ago

True. Manifestos don't have as much impact to change the wider market.

If you ask an exec at a non-tech company about their inspiration for adopting Agile, they'd mention "that Spotify video from 2014".

Agile didn't really take off beyond early adopters until Henrik Kniberg made that Spotify agile video.

 **1**   Reply Share Report Save Follow

Putrid_Acanthaceae · 9 mo. ago

I just spoke to my devops secretary and she said you're wrong op.

 **5**   Reply Share Report Save Follow

poweredbyearlgrey · 9 mo. ago

Really doesn't help when Microsoft release a product called DevOps either.

 **2**   Reply Share Report Save Follow

BeaconRadar · 9 mo. ago

Read Donovan Brown's definition of DevOps

 **1**   Reply Share Report Save Follow

Mentals__ · 9 mo. ago

This is a pretty solid, laid out video on the term and it's evolution

<https://youtu.be/0yWAtQ6wYNM>

 2   Reply Share Report Save Follow

slowclicker · 9 mo. ago

Not the first or last time this will happen. Not the first or last post annoyed about it.

 2   Reply Share Report Save Follow

[deleted] · 9 mo. ago

Honestly this ship has sailed. Like Agile before it, which was a set of practices and beliefs focused on making software developers being more involved with the business so that together they could deliver better solutions, but instead was co-opted by managers (all those stupid certifications and scaled agile frameworks is one example) as a way of controlling all the tiny steps involved in delivering value with software.

DevOps now is mostly a very poorly defined role that means more or less System Administrator but now with more fancy tools and lots of automation.

There are plenty of success

cases of DevOps practices being properly adopted in companies and where People > Processes > Tools actually means something. But in a lot of places, DevOps is seen as a magic bullet just like Scrum was, where people think that if you adopt the processes and tools things will magically fix themselves. The most important part, which is having the right people with the right mindset of continuous feedback, self-improvement and that create and foster an culture of openness, well... that's just too damn hard work, so go back to your Jenkins and fix that pipeline ASAP, some manager just promised something unrealistic to someone and now it's up to you to deliver.

 **2**   Reply Share Report Save Follow

Relevant_Pause_7593 · 9 mo. ago

Why do you think a manifesto will help? No one reads the agile one and look at what a dumpster fire that is in many places.

 **2**   Reply Share Report Save Follow

dataxxx555 · 9 mo. ago

While you wrote this, one of the Terraform-using IAM gatekeepers wrote a new CI yaml and provided value to their internal teams, thus radiating value to customer.

>> We need a manifesto (akin to the 12 Factor App) with a set of rules for how to adopt DevOps into a team and design a system with it in mind.

I think there was a manifesto that was written and updated since 2002 on this precise subject :) Do note, that if you wrote a descriptive manifesto on applying behavioral practices to tech, people will take it as prescriptive and there will be a thread against your manifesto with many of these points in ~14 months

 2   Reply  Share  Report  Save  Follow

homelaborator · 9 mo. ago

It's the same inevitable problem over and over.

Someone comes up with a new abstract system or conceptualisation that they hope can be applied en masse, but there is an ever shrinking number of people who have the capacity or inclination to understand these new

abstractions.

Marketing and grasping managers glob onto the latest buzzwords without knowing what they mean, just knowing that by "embracing" them, they will progress their career or sell more shit or whatever tangential thing they are into.

The regular worker has this "thing" foisted on them by managers who don't know what it is or what they are doing and resist the change which makes no sense.

The manager claims transformative renovation and moves on to greener pastures (or else lets the whole thing die on the vine) and moves onto the next buzzword.

The "one big idea" that DevOps, Agile, Scrum and the broader group offer is to found practice in empiricism, to constantly evaluate and adjust. That way *whatever* brand new idea you adopt, you'll at least know if it's doing *something*.

Naturally, a clever manager might resist measuring and evaluating since they will realise it could show them in a bad light. But, hey, that's just the cost of hierarchical, top down management... but that's a different problem.

↑ 2 ↓  Reply Share Report Save Follow

dossier · 9 mo. ago

For OP, what's your opinion of The Phoenix Project book? I was considering reading that to better understand "real" DevOps

 **2**   Reply  Share  Report  Save  Follow

thinkmassive · 9 mo. ago

Not OP, but I read a fair amount of books in our field. The Phoenix Project provides a very basic concept of DevOps principles. It's a relatively quick and easy read, so it probably won't hurt to start there. Keep in mind it will show its age a bit by now.

For someone who's been working in the industry a couple years, especially if your organization is already practicing DevOps and Agile methodologies (or at least attempting to) then I highly recommend Team Topologies, as I mentioned in [another comment](#)

 **2**   Reply  Share  Report  Save  Follow

[Continue this thread →](#)

MrScotchyScotch **OP** · 9 mo. ago

Honestly? I'm not a big reader, so I put it down after 10 pages and went and looked up blog posts, conferences talks, podcasts, etc, and cobbled together my own notes. There's a huge swath of different concepts that aren't named or explained in the book, and inter-related fields like Lean Manufacturing (DevOps tends borrow a lot from Lean).

 **2**   Reply  Share  Report  Save  Follow

smcarre · 9 mo. ago

People don't get the term

It seems you don't get that term's meanings can change. Welcome to the future, here DevOps stopped being something mentioned by manager books of 2013 and is now an actual thing used by thousands of tech companies.

and getting in the way of progress

How?

I think it's time for it to die.

And then proceeds to defend the "original" meaning of the term. How is it wanting it to die?

We know DevOps is not a role. But of course there are thousands of roles with

"DevOps" in the title.

No, we don't. I **know** it's a role, I can check it in my own job description, it says it right there "DevOps Engineer". It seems you are wrong, DevOps is a role.

We know DevOps is not one thing you "do". But people are constantly asking - or telling - how they "Do DevOps" or "Get Into DevOps" or "Become A DevOps".

Again, we don't. If so many people claim to be doing DevOps, have you considered that **maybe** it is you the one who isn't doing DevOps?

They all think the word means somebody who writes Terraform and gate-keeps the AWS IAM Administrator role. Common use of the term implies it means "setting up server software in Linux".

That's like saying that people think that "Developer" means someone who writes Python and reviews merge requests. Is being a Developer equivalent to writing Python and reviewing merge requests? No. Can writing Python and reviewing merge requests be specific tasks performed by someone that fills the role of a "Developer"? Of course it can, same goes for writing Terraform and gate-keeping AWS IAM Admin roles

for a "DevOps" role.

Add in a few buzzwords like
IaC and Immutable
Infrastructure, and that's all
there is to it.

It seems you also don't know
what the term "buzzword" is if
you think IaC or Immutable
Infrastructure are "buzzwords".
What are other "buzzwords"?
JavaScript? AWS? Git?

It is so completely
misunderstood by 99.99% of
people that almost nobody
uses it in the proper context

At what point does a word stop
being "misunderstood" by the
majority that understands A and
starts being misunderstood by
the minority that understands
B? Pretty sure it is considerably
below 99.99%. "Gay" used to
mean "happy" 70 years ago and
now 99.99% will understand
first "homosexual". Is the 0.01%
wrong? Not necessarily but if
they are talking with anyone in
the other 99.99% of the people
and use the term "gay" and
understand "happy" they likely
are.

The only people that do are
the tiny, tiny few that have
actually read all the books
and blog posts and gone to
the conferences

Ah yes, the actual people with
the highest authority to say
what a tech term means: people
whose work is writing books

and blogs.

If 9999 actual engineers say that DevOps is writing Terraform and 1 blogger says it's shortening the lifecycle and whatnot, then I'm sorry but in the tech industry DevOps means writing Terraform.

Of course, a select few know DevOps' real definition

Ah yes, a select few who can Google "DevOps" and enter the first link. That's a really select group I wish I was part of... oh wait, I was part of that group before I even worked in tech...

Unless you are a consultant working on Digital Transformation, you won't learn what DevOps actually encompasses, and probably will never work on all aspects of it as an IC.

What even entails being "a consultant working on Digital Transformation"? Is a developer writing a test case working in digital transformation (assuming there wasn't a test case before)? You are just replacing a vague term with another vague term.

This perpetually-misunderstood nonsense word will continue to be a blight on the practices it is intended to push

How is this "a blight"? Who suffers from it? Noisy gate-

keeping blogging experts? I'm a DevOps engineer and I don't suffer from this "blight", I even profit from it, I got from a position mostly writing Terraform templates to a position managing pipelines (with a nice pay rise in between) thanks to both being "DevOps" roles.

We need to kill the term, so that new terms that speak to more specific aspects of DevOps can arise. Then it will be easier for people to be aware of them, learn how they work, and try to implement them.

How do you propose to kill the term? What do we win from it? You think people won't start using your new DevOps term for what DevOps means today?

We need to remove the DevOps role and replace it with multiple bodies of knowledge aimed at different existing tech roles.

How do you propose to remove the role? Will you convince the thousands of companies that use the term for their roles?

We also need to avoid roles that merely gatekeep access to production services/accounts, and focus on building the platforms that enable multiple roles to achieve system-level functionality without becoming systems experts.

No, we don't. Gatekeeping is a useful thing in some contexts, managing systems is one of them as it allows some users that lack the knowledge, trust or diligence to work on them without the ability to disrupt them in any significant way. From this sentence alone I'm convinced this rant was sparked because someone denied you a request for an admin access in your work AWS account.

We need to remove the cargo-cult aspects of DevOps buzzwords and develop real engineering disciplines based on DevOps practices

Yes, that's exactly what already happened and you are complaining about. A blogger came and said "DevOps is shortening lifecycles" to an actual engineer, the engineer found out that using GitLab CI shortened the lifecycles so to him writing GitLab pipelines is "doing DevOps" and now you are complaining that the engineer removed his cargo-cult about the term.

We need to show teams real world examples of DevOps culture that achieve both trust and excellence in the production and operation of software systems.

Internet is full of them. Guess what the real world examples are? Writing Terraform and gate keeping AWS roles.

We need a manifesto (akin to the 12 Factor App) with a set of rules for how to adopt DevOps into a team and design a system with it in mind.

They already exist in your myriad of blogs, books and conferences. Nobody gives a damn because they are either worthless or too vague and apply to everything that is already DevOps.

We've gotta stop calling it DevOps first, though... so we need new ways to refer to those specific components of DevOps without using the word "DevOps".

You haven't addressed a single reason to do any of that beyond you not liking people referring to DevOps as a role. What do we achieve with that? What problems do we avoid?

0/10, this post fell right in the fence between "this has to be pasta" and "this isn't pasta". BRB, gonna change my LinkedIn title to "YAML developer".

 **2**   Reply  Share  Report  Save  Follow

darkn3rd · 8 mo. ago ·
edited 8 mo. ago

This is not just DevOps, but other domains in tech this happens as well. At first you get the practitioners that use it and embrace it, but then comes

people outside of technology writing about DevOps, then there comes the marketing that tries to "sell" DevOps, then there comes the textbook writers that use the marketing to document what DevOps is, and next thing you have students in colleges reading about DevOps in these textbooks and proud professors talking about it with authority as if they were the implementer of DevOps.

There has been efforts like CALM, CALMS, and CALMR:

- <https://blog.sonatype.com/principle-based-devops-frameworks-calms>
- <https://www.atlassian.com/devops/frameworks/calms-framework>
- <https://www.scaledagileframework.com/calmr/>

There's also initiatives to create a DevOps Maturity Model.

- <https://newrelic.com/resources/ebooks/devops-maturity-phases>
- <https://github.com/joelparkershenderson/maturity-models/blob/main/examples/agile/agile-devops-maturity-model-by-hewlett-packard-enterprise-hpe/index.md>
- <https://github.com/adidas/adidas-devops-maturity-framework/blob/master/fr>

[amework/devops_maturity_framework.md](#)

Atlassian used to have some good material, but now searches lead to popup surveys without literature.

Searching for information is really hard, as results are heavily marketing-sales laden with sources attempting to SELL you something, rather than real practitioners. I think this domain would benefit to a DevOps Book of Knowledge, similar to [PMBok](#) and [BABok](#).

A good source of knowledge on state of Devops is DORA - <https://www.devops-research.com/research.html>

 **2**   Reply  Share  Report  Save  Follow

jeerabiscuit · 9 mo. ago

How about system engineering?
It already is a concept

 **5**   Reply  Share  Report  Save  Follow

MrScotchyScotch **OP** · 9 mo. ago

Sure, I think Systems Engineering is a great discipline to focus on. But I'm looking for a way to say, "this is the DevOps way to do Systems Engineering", and mark out those specific differences, and how other roles can/should interface with SysEng in a DevOps way. Some word(s) to express a specific *way* to do that traditional role. Almost like, a "DO Systems Engineer", a "DO Developer", "DO QE", etc.

 **1**   Reply  Share  Report  Save  Follow

[deleted] · 9 mo. ago

I think it's the term's "fault": DevOps...? Aaaah, so Dev + Ops in one person, gotcha! Also I applied to a few "devops" positions, and I can safely say nobody even agrees on the misunderstood definition either :D

 **4**   Reply  Share  Report  Save  Follow

GeorgeRNorfolk · 9 mo. ago

a combination of specific practices, culture change, and tools intended to shorten SDLC by reducing the time between committing a change to a system and it being continuously delivered into normal production while ensuring high quality and reliability

Where did this definition come from?

 **3**   Reply Share Report Save Follow

MrScotchyScotch **OP** · 9 mo. ago

Honestly I cribbed it from [Wikipedia](#). It's a mashup of two similar definitions that encompasses most of the concept of DevOps.

 **1**   Reply Share Report Save Follow

DirtyDizzal19 · 9 mo. ago

I'm a software engineer who does devops. Devops is developer operations, like any other operation role I'm here to support the developers/engineers make their day-to-day lives easier. Typically that is through setting up pipelines, building out infrastructure as code, building better local development environments, documentation, SDLC improvements, building internal services/apis, and general automation. I have held four different devops positions and this is how I explain it every time to the team.

 4   Reply Share Report Save Follow

Zauxst · 9 mo. ago

I started as a system administrator that was highly interested in scripting and automation...

I guess that was enough to qualify myself for devops.

This came with a huge pay increase and I'll take it... But god damn I hate it whenever I talk to people and have to explain that devops as a title is stupid.

 3   Reply Share Report Save Follow

ipaqmaster · 9 mo. ago

Uh... yeah what's the point of this post? This hasn't taught me how to be 1x debops at all. /s



2



Reply

Share

Report

Save

Follow