

Breaking DevOps

Max Körbächer @ CodeRepublic

Stay connected

maxkoerbaecher; mkoerbi; mkorbi



Co-Founder & Sr. Manager Cloud Native Engineering
@ Liquid Reply

Kubernetes Release Team

Cloud Native Enthusiast and Advocate





Why DevOps currently is broken

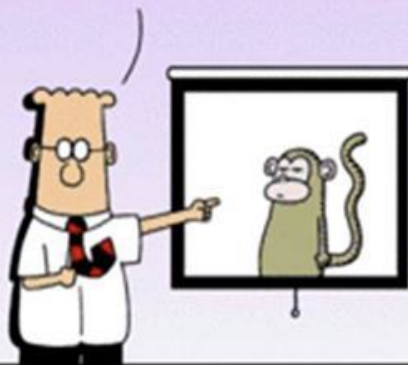


IF WE MIGRATE THE
ENTERPRISE TO CLOUD,
USE ANALYTICS, IOT AND
DEVOPS FOR ALL
SERVICES,



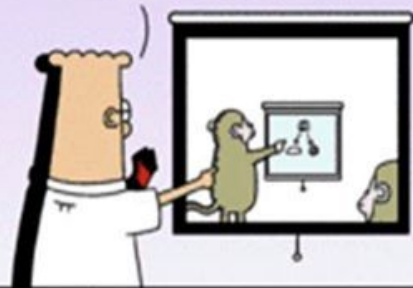
Dilbert.com DilbertCartoonist@gmail.com

THE ENTIRE COMPANY
CAN BE MANAGED BY
ONE MONKEY.



1-18-10 ©2010 Scott Adams, Inc./Dist. by UFS, Inc.

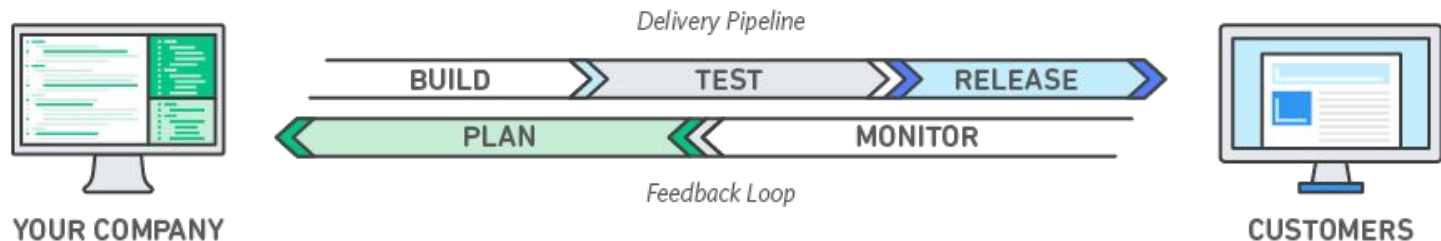
PLUS A SECOND
MONKEY TO LOOK AT
THE POWERPOINT
SLIDES FROM THE
FIRST MONKEY.



What is DevOps?

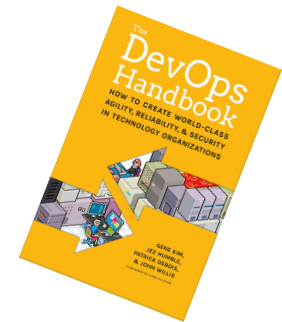
Blogs & Articles

- **Infrastructure Automation** – create your systems, OS configs, and app deployments as code.
- **Continuous Delivery** – build, test, deploy your apps in a fast and automated manner.
- **Site Reliability Engineering** – operate your systems; monitoring and orchestration, sure, but also designing for operability in the first place.



What is DevOps?

The Phoenix Project



The Principles of Flow

Make **your work visible**; Limit work in process (WIP); Reduce batch sizes; Reduce the number of handoffs; Continually **identify and address your bottlenecks**; Eliminate hardships and waste in the value stream

The Principles of Feedback

Design a **safe system of work**; See problems as they occur; Swam and solve problems to build new knowledge; Keep pushing **quality closer to the source**; Enable optimizing for downstream teams

The Principles of Continual Learning & Experimentation

Enable an **organizational learning** and safety culture; Institutionalize the improvement of daily work; Transform **local discoveries into global improvements**; Inject resilience patterns into our daily work; Leaders reinforce a learning culture



What is DevOps?

DevOps

From Wikipedia, the free encyclopedia



Some of this article's **listed sources may not be reliable**. Please help this article by looking for better, more reliable sources. Unreliable citations may be challenged or deleted. *(December 2018)* *(Learn how and when to remove this template message)*

DevOps is a set of practices that combines **software development** (*Dev*) and **IT operations** (*Ops*). It aims to shorten the **systems development life cycle** and provide **continuous delivery** with high **software quality**.^{[1][2]} DevOps is complementary with **Agile software development**; several DevOps aspects came from Agile methodology.

Contents [hide]

- 1 Definition
- 2 History
- 3 Toolchains
- 4 Relationship to other approaches
 - 4.1 Agile
 - 4.2 ArchOps
 - 4.3 TestOps
 - 4.4 Continuous delivery
 - 4.5 DataOps
 - 4.6 Site-reliability engineering
 - 4.7 Systems administration
 - 4.8 WinOps
 - 4.9 Toyota production system, lean thinking, kaizen
 - 4.10 DevSecOps, Shifting Security Left
- 5 Goals
- 6 Criticism
- 7 Cultural change
 - 7.1 DevOps as a job title
 - 7.2 Building a DevOps culture
- 8 Deployment
 - 8.1 Architecturally significant requirements
 - 8.2 Microservices
 - 8.3 DevOps automation
- 9 Adoption
 - 9.1 DevOps practices and adoption
- 10 See also
- 11 Notes
- 12 References
- 13 Further reading

Software development

Core activities

Processes · Requirements · Design · Engineering · Construction · Testing · Debugging · Deployment · Maintenance

Paradigms and models

Agile · Cleanroom · Incremental · Prototyping · Spiral · V model · Waterfall

Methodologies and frameworks

ASD · **DevOps** · DAD · DSDM · FDD · IID · Kanban · Lean SD · LeSS · MDD · MSF · PSP · RAD · RUP · SAFe · Scrum · SEMAT · TSP · OpenUP · UP · XP

Supporting disciplines

Configuration management · Documentation · Software quality assurance (SQA) · Project management · User experience

Practices

ATDD · BDD · CCO · CI · CD · DDD · PP · SBE · Stand-up · TDD

Tools

Compiler · Debugger · Profiler · GUI designer · Modeling · IDE · Build automation · Release automation · Infrastructure as code · Testing

Standards and Bodies of Knowledge

BABOK · CMMI · IEEE standards · ISO 9001 · ISO/IEC standards · PMBOK · SWEBOK · ITIL · IREB

Glossaries

Artificial intelligence · Computer science · Electrical and electronics engineering

Outlines

Outline of software development

V · T · E



Everything clear now?



Nothing is clear

companies all do it differently

developer should become operations

operations should do development

“it is just IT, you should know it” - right, and your Orthopedist does also do
brain surgeries



DevOps appear when IT was stressed

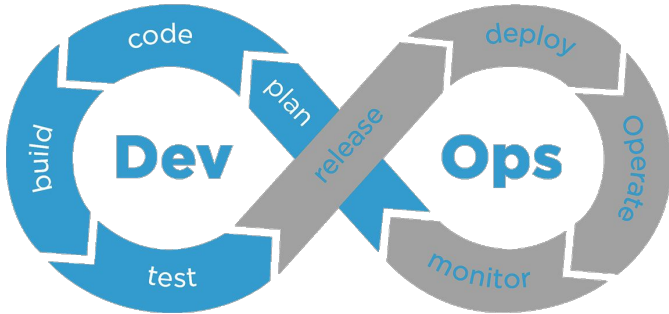
Remember?

- When you need to order new VMs (yes, order!)
- Developer complain about that the Operations doesn't understand how to run the App
- Operations complain about that Developer doesn't know how to write a good App
- Everyone else complains in general about IT and its costs
- Microservices everywhere (in the mouth), CI/CD tools hit the adoption, business talking about bringing features faster to the client
- Less professionals on the market



DevOps worked well

and works well, but not for any case



It works when:

- you build it you run it (partially) is applicable
- system complexity can be explained on a single piece of paper
- vertical, lean responsibility
- your team member can take over also other domains tasks (not as expert, but to keep alive)



Moving IT

Why DevOps (as we know it) is just an interim solution



Filling the Gap

Development + Operations

Then



???

Now/Next

front & back end + DB

architecture

many services, many
storages, multi device

2-4 major, 4-8 minor

releases remain

once or twice per day

commercial, one tech
stack, stability

design principles

best fit, flexibility, open
source, exchangeable

unit + with the release

test complexity

on every commit from
unit to functional tests

one version, one API

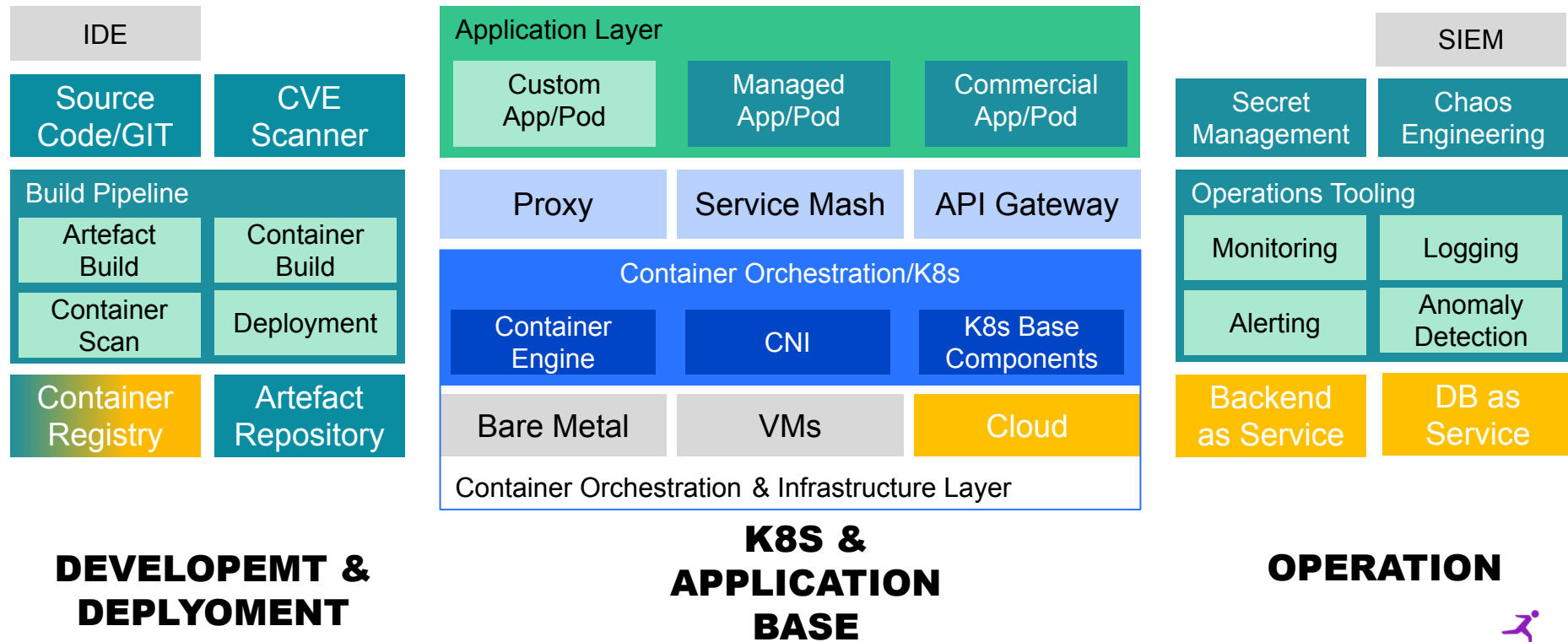
compatibility

multiple version,
multiple API versions

DevOps

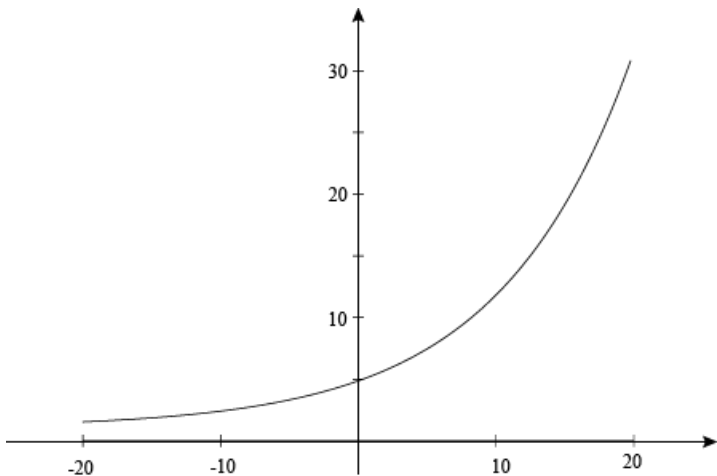


Don't take this slide too serious, but a DevOps Team which should focus on a vertical capability can't cover all aspects



System complexity is growing

for the good and the bad



With growing system complexity, the underlying idea and approach of devops can't fulfill the needs.



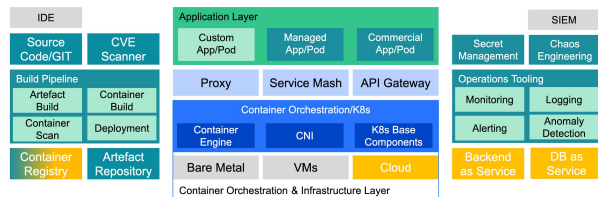
Today's platforms are more, deeper

Today platforms and systems have to integrate in a variety of other platforms and systems.

While many of them are maintained, you still need to know how to use, build (on) and utilize them.

What is often misunderstood, every topic is so much more today:

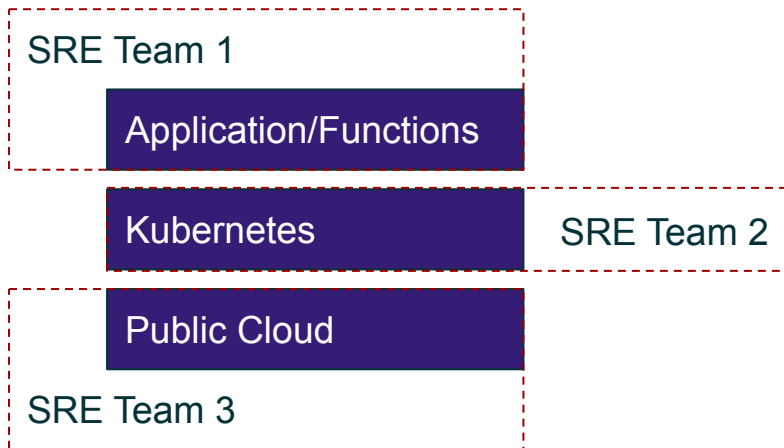
- K8s, Container
- Serverless, ICP
- Cloud, Hybrid, Edge
- Monitoring, Logging, Tracing & Observability
- CI/CD, GitOps, Release Mgmt., Lifecycle
- Security, Hardening, Scanning, Intrusion detection





~~DevOps~~ everyone and none

Site Reliability Engineering -> Gives the purpose to the Job



Broaden the expected responsibilities and field of knowledge is valuable as well as actively maintaining feedback loops.

But I believe tall vertical towers doesn't deliver faster, high quality well designed solutions.

Except you have big team which can cover all aspects.

For many SRE is just another name of DevOps, in fact it is very clear described what is the purpose of SRE and how achieve a well SREism

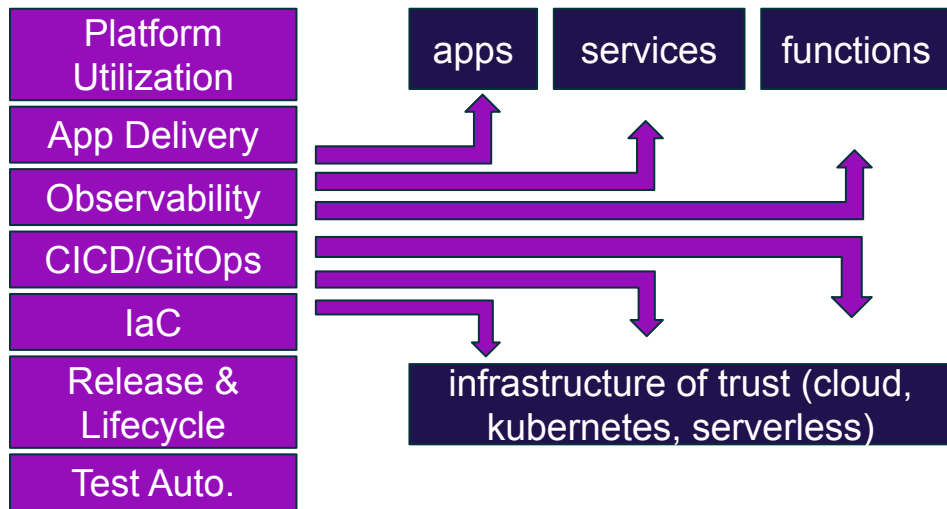


And where did DevOps go?

DevOps makes Development working by supporting the integration

DevOps fills again a gap:

- how to run on public cloud?
- how to run on container?
- what I need to consider from Kubernetes?
- how to control versions & releases?
- what is best to use to develop and deliver an app?
- what I don't have to reinvent?
- ...



We just start to enter a new complexity

With Public Cloud many things got easier, but the amount of available services is overwhelming.

Moving on to a Kubernetes based system brings new opportunities but even more complexity. And companies just adopting it more and more.

**Kubernetes is
the Go To
Platform for the
next years**



Where DevOps should be

apps

services

functions

DevOps brings specific knowledge to integrate and run applications on a target platform.

DevOps is an advisor for platform admins and application developers, acting as a mediation point.

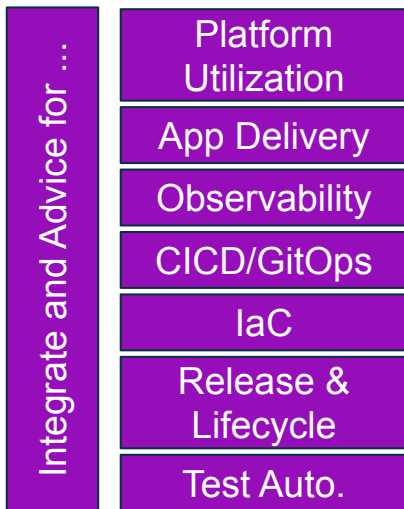
DevOps = Integration Engineer & Architect

infrastructure of trust (cloud,
kubernetes, serverless)



Making the implicit explicit

Implicitly the vast majority of DevOps role descriptions includes our hot topics



So with just excluding the other 50% we have a nearly good role description for an DevOps



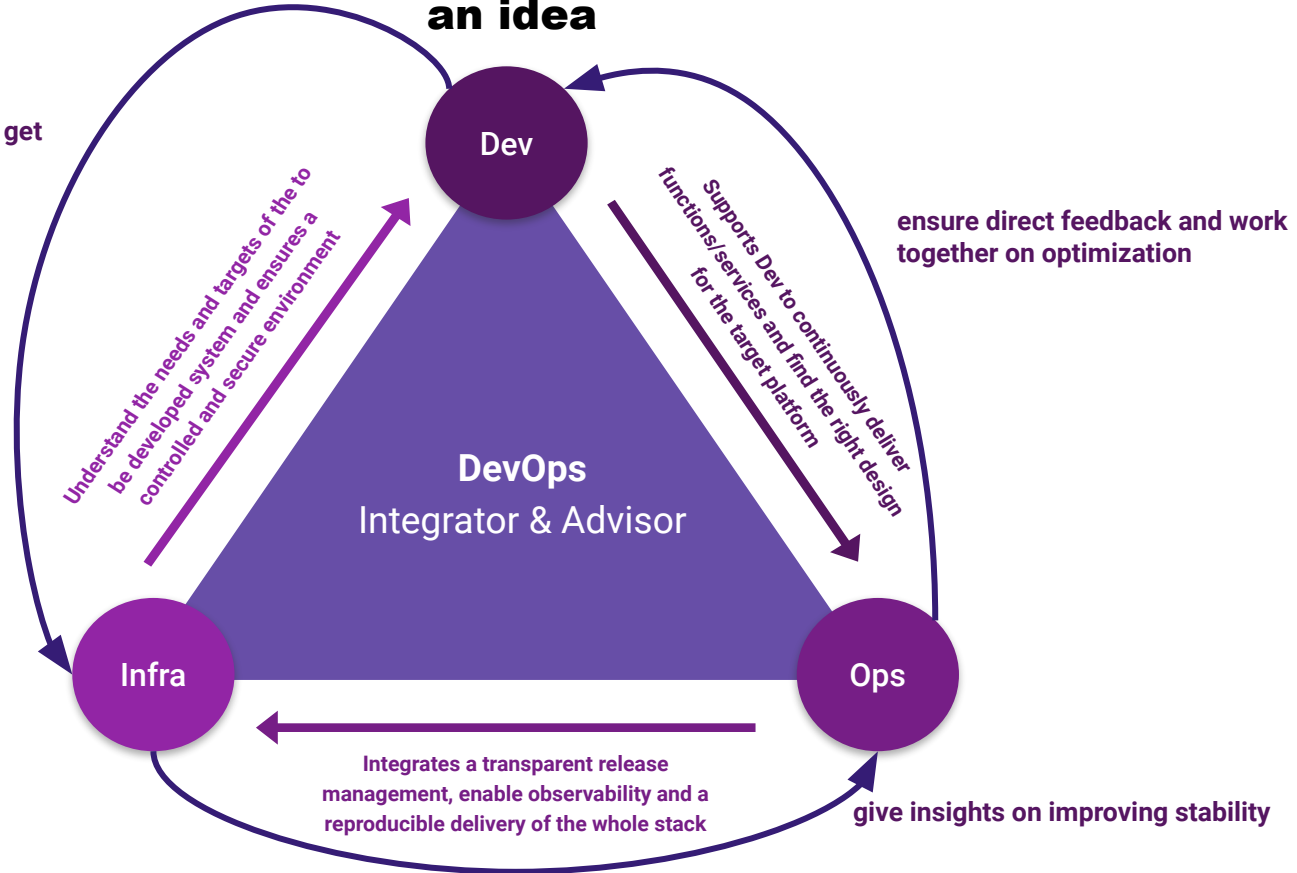
We just have to add some words to make it explicitly clear and than we can start the major challenge to live this role!



The future teams

an idea

communicate your needs and get advice for options



Thank you!

**What will be
your DevOps
journey?**



