



The future of DevOps



Sasha Rosenbaum
@DivineOps





Sasha Rosenbaum
Red Hat

Israeli Air Force

Defense Industry R&D

Cloud Consulting

Microsoft

GitHub

DevOpsDays Chicago since 2014

@DivineOps

How about you?





The Past

Technology

1990s:

Getting a new
server for an
application:
2-3 months



Backup





William Herold

@willigula



Replying to @DivineOps

I don't know, but I have misguided nostalgia for getting a page & driving to our data center to physically restart a server in the middle of the night.

Date	Release name
1990	SQL Server 1.1 (16-bit)
1992	SQL Server 4.2A
1993	SQL Server 4.21a
1995	SQL Server 6.0
1996	SQL Server 6.5
1998	SQL Server 7.0
2000	SQL Server 2000
2003	SQL Server 2000 64-bit
2005	SQL Server 2005
2008	SQL Server 2008
2010	Azure SQL database

Software release
cadence:
2-3-year cycle

Merge hell

Merging the development
branches and completing the test
procedures could take months

Maintenance
windows



**System
Downtime**

**Saturday, June 4 -
Sunday, June 5**

Expected Availability

< 99%

3.65 days / year

Unavailable systems were estimated to have cost American businesses \$4.54 billion in 1996.

Culture



© 2018 Forrest Brazeal

"I knew this organization was full of siloes.
I just didn't think they'd be so heavily defended!"

Traditional
IT



dev



ops

wall of
confusion

Speed

Reliability

The problem isn't technical.

The problem isn't people.

The problem is socio-technical.



Darmok and Jalad
at Tanagra



Patrick and Andrew
at Agile TO 2008

10 deploys per day: Dev and Ops collaboration at Flickr



Velocity 09: John Allspaw and Paul Hammond

Agile Infrastructure



Velocity 09: Andrew Clay Shafer



Andrew Clay Shafer 雷启理

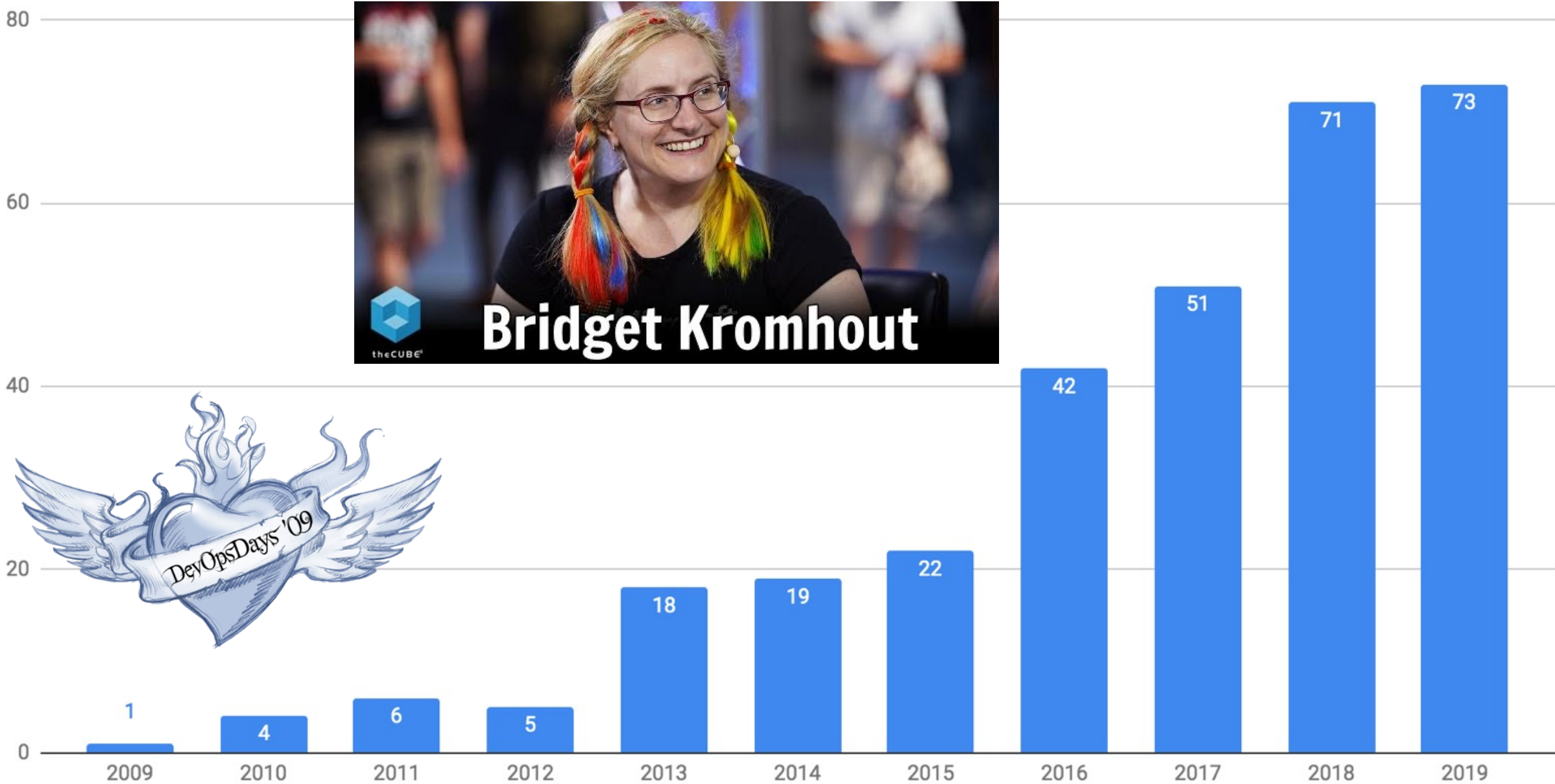
@littleidea

Don't just say 'no', you aren't respecting other people's problems... [#velocityconf](#)
[#devops](#) [#workingtogether](#)



DevOpsDays Ghent 2009: Patrick Debois

devopsdays events



Speed

Reliability



Charity Majors @mipsytipsey · Mar 12



It is a deep seated biological impulse to slow down when we feel cautious. For humans, slow is safety.

Problem is, software physics are different. Speed is safety. It's more like ice skating, or riding a bicycle: the slower you go the more dangerously you wobble.

Charity Majors, CEO of Honeycomb



ELITE PERFORMERS

Comparing the elite group against the low performers, we find that elite performers have...



208
TIMES MORE
frequent code deployments

106
TIMES FASTER
lead time from
commit to deploy



2,604
TIMES FASTER
time to recover from incidents

7
TIMES LOWER
change failure rate
(changes are $\frac{1}{7}$ as likely to fail)



Throughput Stability



Nicole Forsgren. State of DevOps Report 2019

Speed



Reliability

Software delivery is like a muscle.

The more you use it, the stronger it gets.

Software Services



Needs to be Operated

Platform Services



Needs to be Operated

Infrastructure Services



Needs to be Operated

In the beginning...

Deployment Checklists

Deployment Plan Template: Blue Theme

Deployment Plan Template - Blue Theme - Coverpage Sheet - Word

FILE HOME INSERT DESIGN PAGE LAYOUT REFERENCES MAILINGS REVIEW VIEW DEVELOPER

Ivan Walsh

Table of Contents

1 Introduction	6
1.1 Purpose	6
1.2 Objectives	7
1.3 Scope of Work	7
1.4 Sites/Sites/LOCATIONS	7
1.5 ASSUMPTIONS	7
1.6 ISSUES	7
1.7 CONSTRAINTS	8
1.8 DEMANDS	8
2 Site Information	9
2.1 Site Diagram	9
2.2 SYSTEM OVERVIEW	9
2.3 Site Preparation	9
2.4 Site Readiness	9
3 Roles & Responsibilities	11
3.1 PROJECT MANAGER	11
3.2 SYSTEM INTEGRATOR	12
3.3 INSTALLATION MANAGER	12
3.4 Chief Technology Officer	13
3.5 Hardware Installation Engineer	13
3.6 Software Support Engineer	15
3.7 CONSULTANTS / SUBCONTRACTORS / THIRD PARTIES	15
3.8 POINTS OF CONTACT	15
4 Schedule	17
4.1 RESOURCE ALLOCATION CHART	18
5 Resources	19
5.1 FACILITIES	19
5.2 HARDWARE	19
5.3 SOFTWARE	19

Copyright 2018. All rights reserved. Page 1 of 30

5.4 Documentation

6.1 Help Desk	21
6.2 Director	21
6.3 SERVICES	21
6.4 NETWORK / VOICE / TELECOMMUNICATIONS	21
7 Training	22
7.1 TRAINING	22
8 Installation Considerations	24
8.1 SAFETY	24
8.2 CODE & INDUSTRY STANDARDS	24
8.3 INSTALLATION PLANNING	24
8.4 WORKAREA & PRODUCTIVITY	25
8.5 INSTALLATION SITE MAINTENANCE	25
8.6 INSTALLATION RECORDS	25
9 Installation Process	26
9.1 PREPARING	26
9.2 INSTALLING	27
9.3 TESTING	29
9.4 SITE MONITORING	29
10 Transfer to Operations	31
10.1 RESOURCES	31
10.2 PROCEDURES	31
10.3 CONFLICTS Areas of Change	31
10.4 Transition Activities	31
10.5 MONITORING	31
10.6 Help Desk	31
10.7 NOTIFICATION OF DEPLOYMENT	31
11 Appendix	32
11.1 ACRONYMS AND ABBREVIATIONS	32

Copyright 2018. All rights reserved. Page 1 of 30

1 Introduction

This Deployment Plan is a "how-to" guide to implement a solution into production. It either works to deploy an IT system, technical solution, or equipment from its current environment into a new environment. It provides detailed deployment guidelines and helps drive the deployment phases. In general, this involves preparation, validation, testing, verification and hardware. Introduce the Deployment Plan by summarizing the high-level activities necessary to get the target system into a production environment. Depending on the scope of the deployment, this document will describe the installation, configuration, operational and business processes that will be modified as a result of the deployment process. As not all sections may be relevant to your project, please adjust the template to match your needs.

1.1 Purpose

Identify the purpose of the Deployment Plan and its intended audience, which is typically a combination of client representatives and members of the system integrator's team. Sample text:

This document provides guidelines for the creation of the Deployment Plan by the [System Integrator]. The purpose of the Deployment Plan is to describe the factors necessary for a smooth deployment and transition to operations. This covers the tasks involved in preparing, installing, training, stabilizing, and transferring the solution to daily operations as well as details on different installation scenarios, monitoring for stability, and verifying the reliability of the implemented solution.

The Deployment Plan shall include, at a minimum, a comprehensive Installation Plan that includes a detailed Schedule, Training Plan, and Safety Plan.

The following sequence of deployment activities will be performed by the [System Integrator]:

- Provide a Deployment Plan Schedule showing the individual tasks associated with the installation of equipment/software/hardware etc.
- Pre-Installation activities include working with the Client to finalize the Installation Plan, Training Plan, Communication Plan, Schedule and other deployment documents.
- Ensure that all safety procedures are in place.
- Verify that the equipment and software is installed properly.
- Verify that each subsystem communicates properly.
- Verify that all installed equipment and software operates properly by conducting systems testing.

The System (System Integrator) shall provide clearly defined equipment and software deployment techniques that will be used in the Installation Plan and, if accepted by the [Client], applied to all the equipment, subsystem, and software installations.

Copyright 2018. All rights reserved. Page 1 of 30

2 Site Information

Place the deployment site in context by describing how the target system will reside in the deployed site. Diagrams and reference to technical specifications and other relevant documentation can be referenced in this section.

2.1 Site Diagram

Ref #	Issue	Impact
#1	Describe the issue	Discuss its effect on the deployment plan
#2	Describe the issue	Discuss its effect on the deployment plan
#3	State "none" if appropriate	

Copyright 2018. All rights reserved. Page 1 of 30

PAGE 1 OF 32 5045 WORDS ENGLISH (UNITED STATES) 57%

Scripts

OS-level APIs



PowerShell

(Windows) configuration
management framework
and scripting language



Jeffrey Snover, 2006

Source Control for Ops

GitHub launch: 2008

Distributed version control
+
Pull Request system
=
Global collaboration

Infrastructure-level APIs

Amazon Web Services: 2002

Amazon Cloud Computing: 2006

Darwinian Pressure

The new models evolved
due to pressure to deliver
adaptable services at scale.

Netflix, Amazon, Google,
and every 'cloud native'
company built a platform

Because they had to...

'Cloud' evolved from lessons
learned building and operating
these internal services



Infrastructure as code

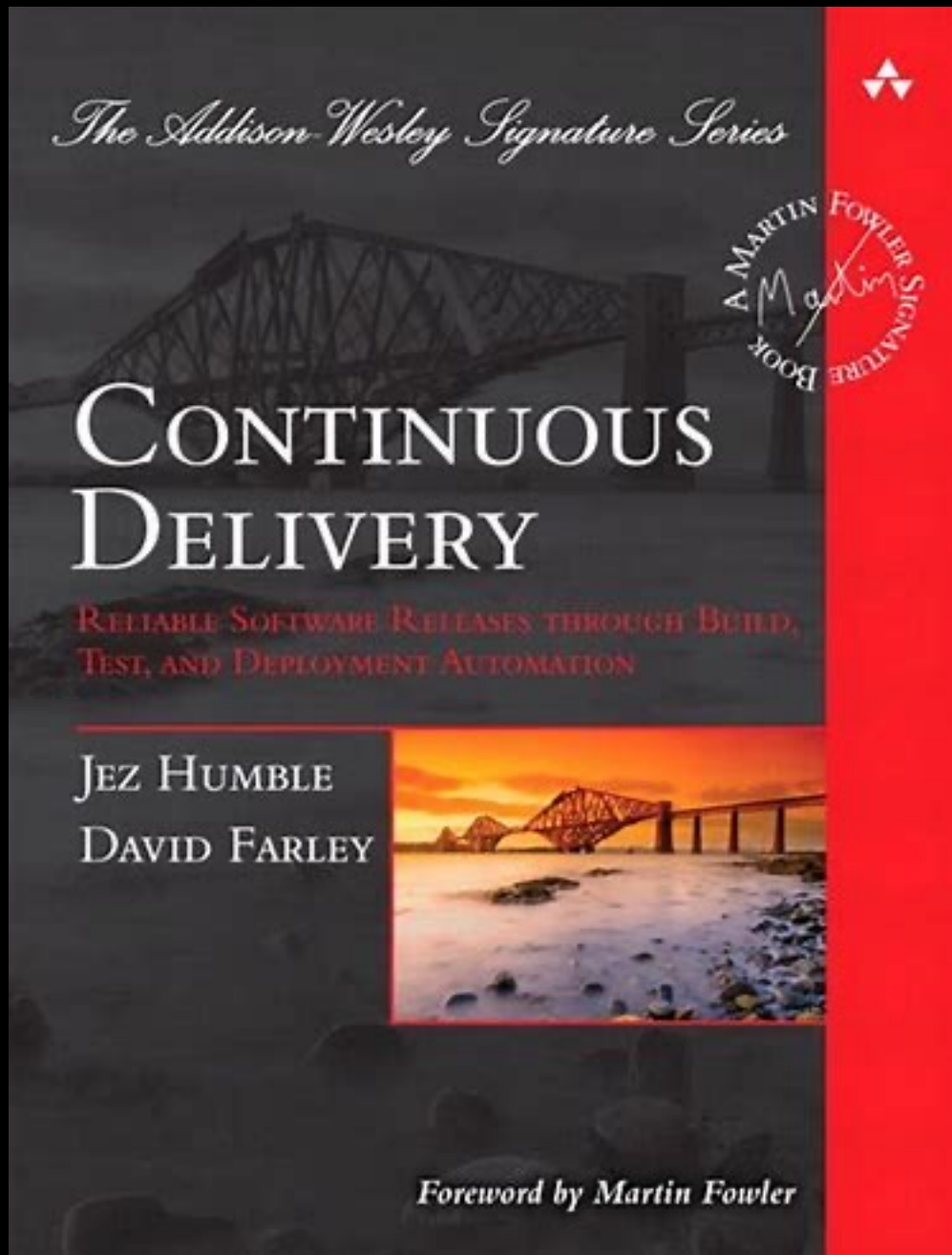


Configuration management
minimizes manual toil and
infrastructure configuration drift

"The traditional model is that you take your software to the wall that separates development and operations, and throw it over and then forget about it. Not at Amazon. **You build it, you run it.**

This brings developers into contact with the day-to-day operation of their software. It also brings them into day-to-day contact with the customer. This customer feedback loop is essential for improving the quality of the service."

–Werner Vogels, CTO Amazon



Jez Humble and Dave Farley: 2010

Continuous Integration (CI)

The process of automating the build and testing of code every time a team member commits changes to version control.

Continuous Delivery (CD)

The approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time.

Software Services



Needs to be Operated

Platform Services



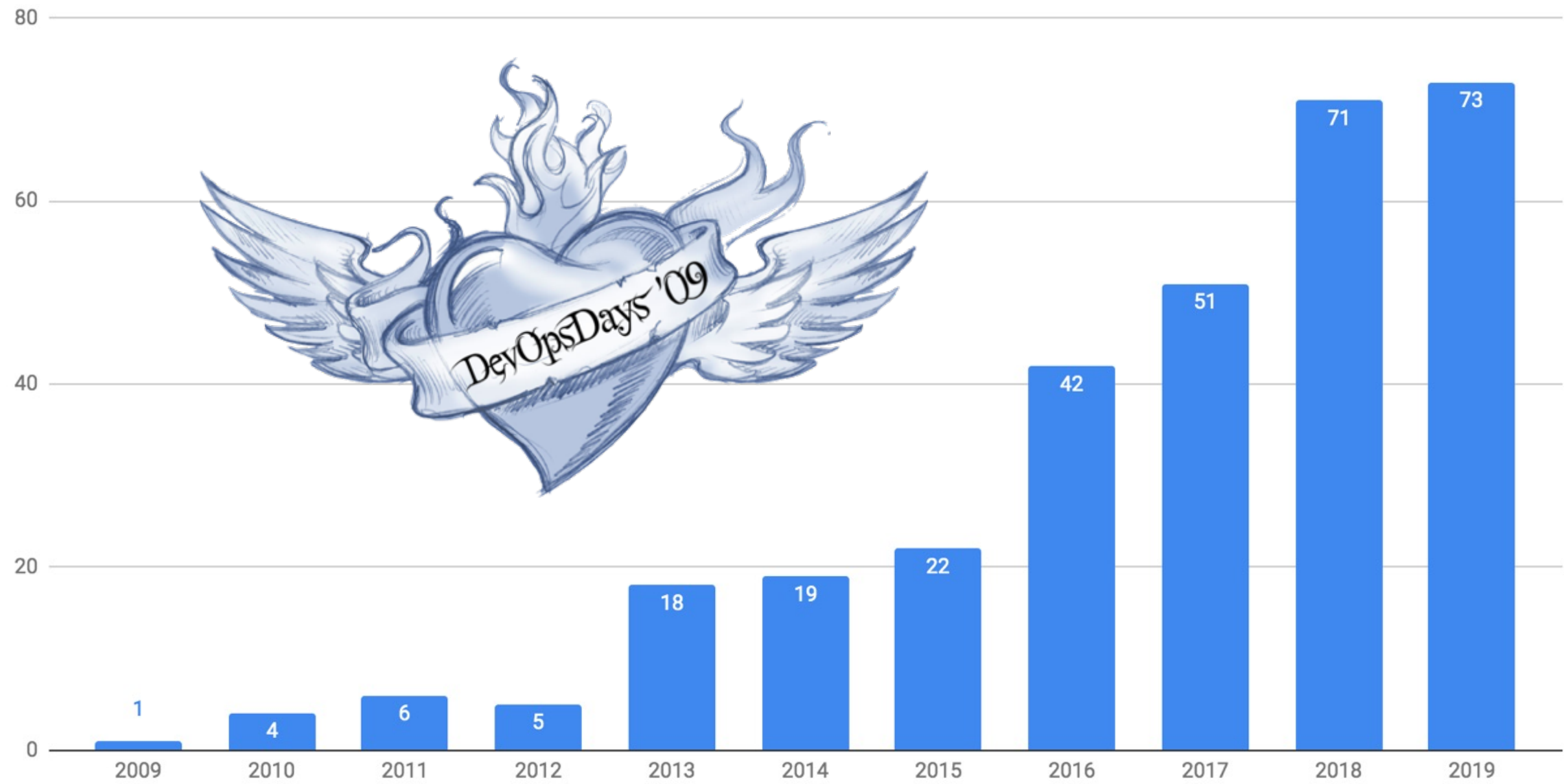
Needs to be Operated

Infrastructure Services



Needs to be Operated

devopsdays events



You will automate me out of a job!

Toil



Toil is the kind of work that tends to be **manual**, **repetitive**, automatable, tactical, devoid of enduring value, and that **scales linearly** as a service grows.

We would not be able to achieve the availability, reliability and speed we have today without automation



Senior Oops Engineer @ReinH · Jul 8, 2020



I mean yes automation eliminates human errors in the sense that those errors will now be performed by machines



20



138



559



Niels Albers

@nralbers



Replying to @ReinH

Automation: How to make mistakes consistently, repeatably and really fast.



The problem isn't technical.

The problem isn't people.

The problem is socio-technical.



The Present

The future is already here.

It's just not evenly distributed

~ William Gibson

DevOps across Microsoft

 <http://aka.ms/DevOps-Stories>

105K

Engineers use the
DevOps platform

4.4M

Builds per month

5M

Work items
viewed per day

2M

Git commits
per month

500M

Test executions
per day

500K

Work items
updated per day

85,000

Deployments per day



bob @rjw1 · May 10, 2018



Repeat after me, **DevOps is a culture not a team or a job title.**



DigitalOcean  @digitalocean · May 8, 2018

"Despite not having a dedicated DevOps team, our engineers developed an auto-devops workflow"



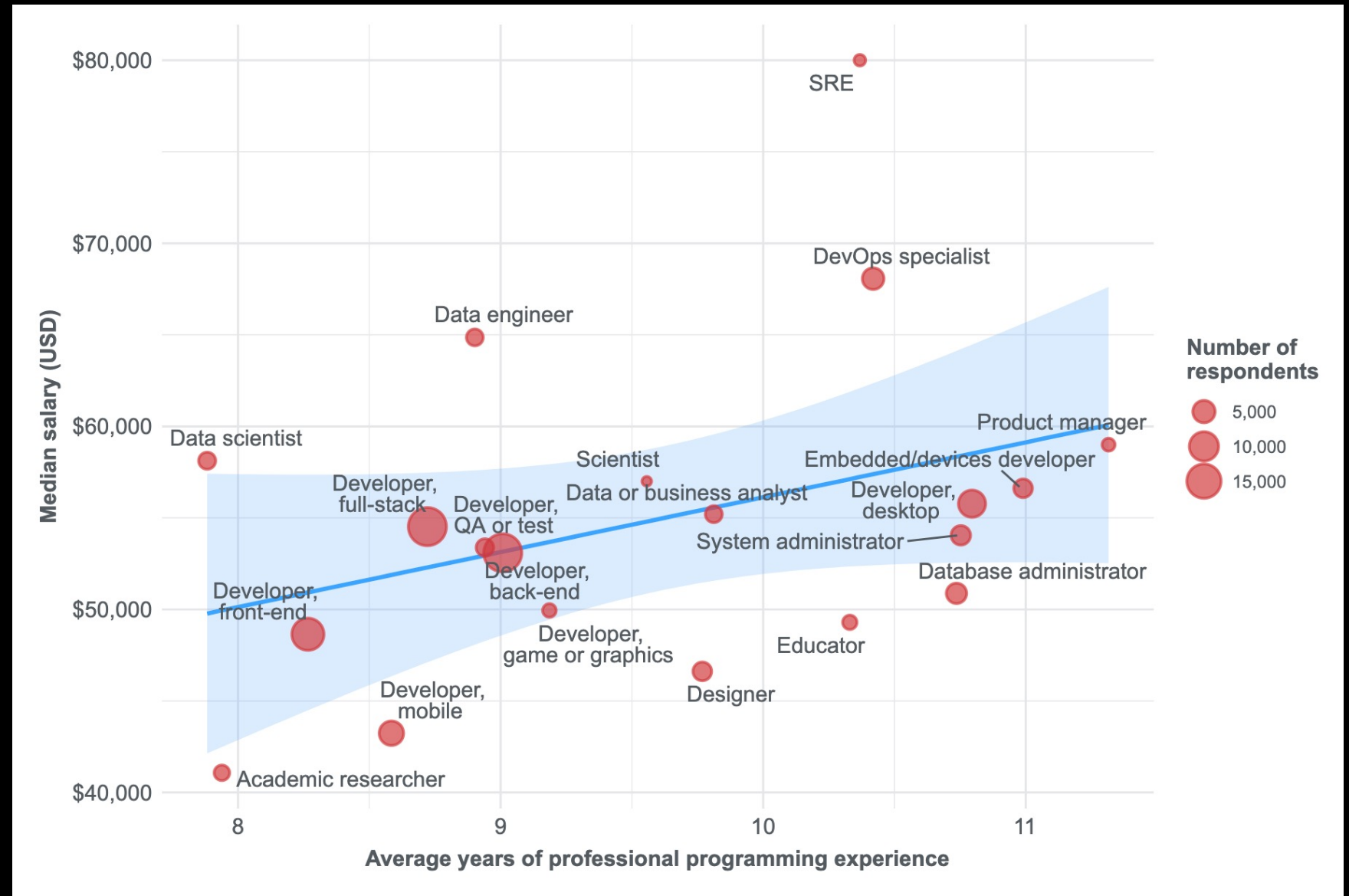
Ahmad
@ahmadjee



Open Space talk on [#talkpay](#) at [#devopsdays](#) [#Austin](#)



DevOps and SRE engineers command a higher salary



Source: Stack Overflow Developer Survey, 2020

We've created new jobs

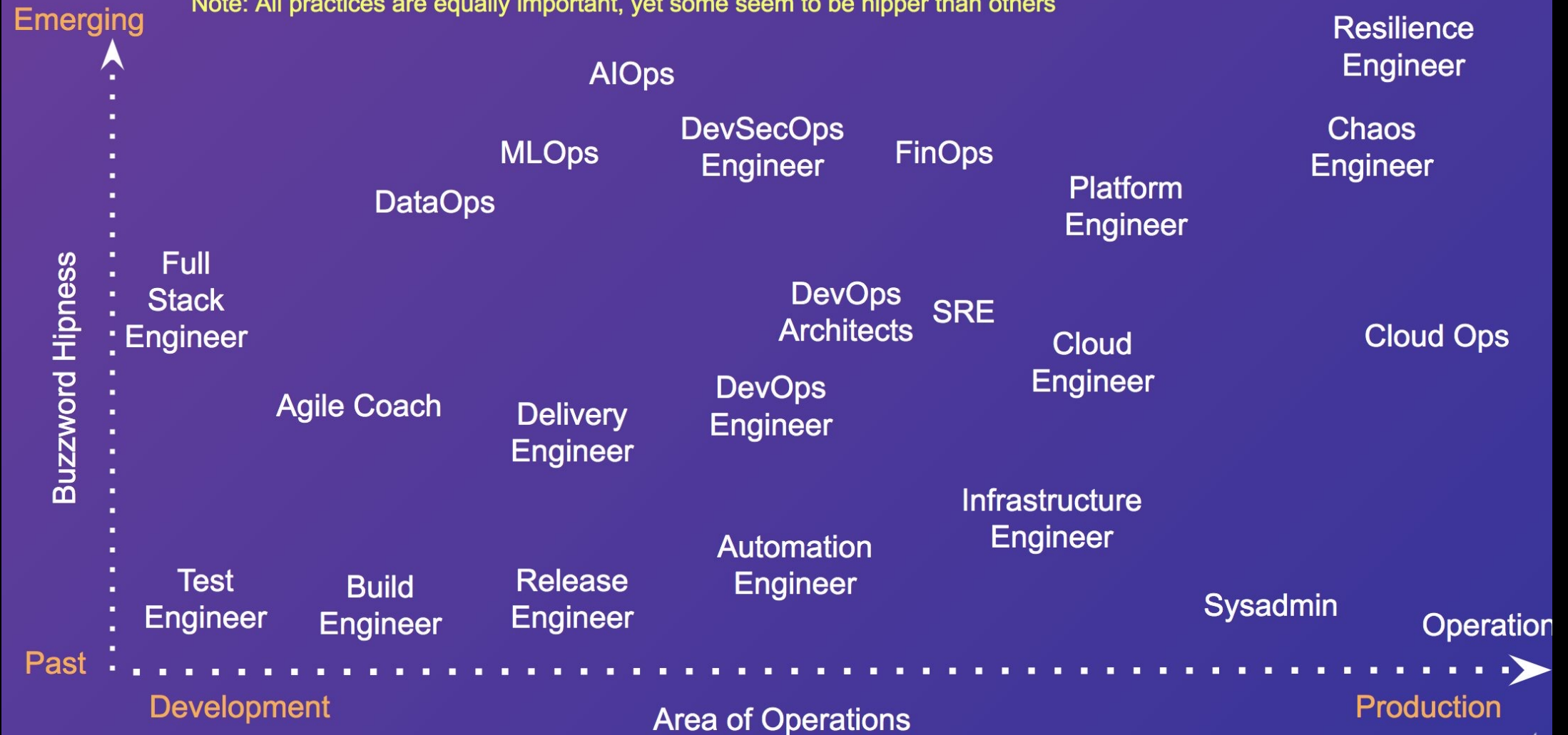
- _ (' ヅ) _ / -

The new jobs got people higher salaries and more interesting work!



Shades of DevOps Industry Terms

Note: All practices are equally important, yet some seem to be hipper than others



We've created new disciplines!

SRE

SRE \simeq Google's DevOps implementation

100% reliability is unattainable

Availability

99.999%

5.26 mins / year

How much does that cost?

Error Budgets

An acceptable level of unreliability

It's a **budget**. It can be **allocated**.

SLI, SLO, and SLA

Service Level Terminology

Describe the metrics that matter, the values we want, and how we will react



Indicators

Defined measurement of an aspect of a service.



Objectives

Target value (or range of values) as measured by an SLI



Agreements

Explicit or implicit contract with users or customers, with consequences of meeting or missing objectives

Monitoring

“Monitoring is how you manage your **known-unknowns**, which involves checking values for predefined thresholds, creating actionable alerts and runbooks and so forth.”



Charity Majors, CEO, Honeycomb

Without **monitoring**, you have no way to tell whether the service is even working

Observability

“Observability is how you handle **unknown-unknowns**, by instrumenting your code and capturing the right level of detail that lets you answer any question ...”



Charity Majors, CEO, Honeycomb

All of this requires collecting and analyzing massive amounts of data

"If we have data, let's look at data. If all we have are opinions, let's go with mine"

- Jim Barksdale



Infrastructure as code



?

MLOps

MLOps

- Massive amounts of data
- Data model versioning
- Model re-use
- Model decay over time
- Compliance considerations

Chaos Engineering

Chaos engineering

The discipline of **experimenting** on a software system in **production** in order to build confidence in the system's capability to withstand turbulent and unexpected conditions.



WAIT...

WHAT?!

makeameme.org

Everybody tests in production

Incident Response

Blameless postmortems

There is no root cause

DevSecOps

More code = more problems

Cyber attacks are
at all time high



Security must be an integral part of
the software development lifecycle

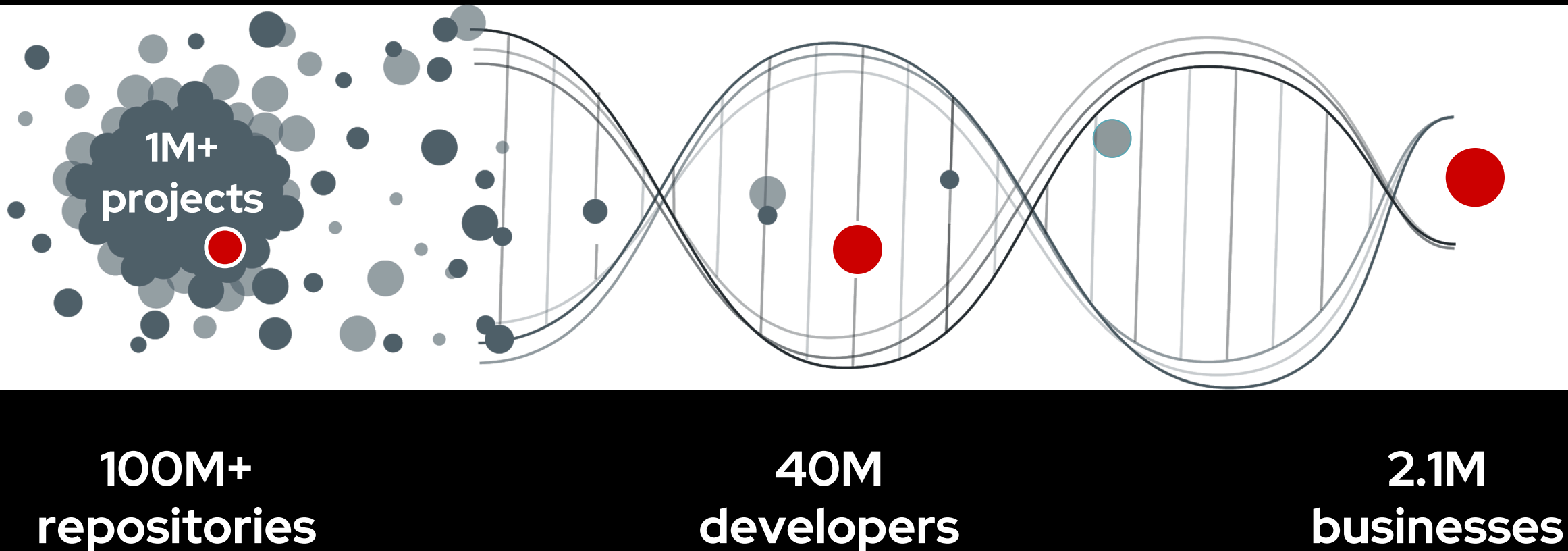
There is so much more...

The industry has evolved

Open Source

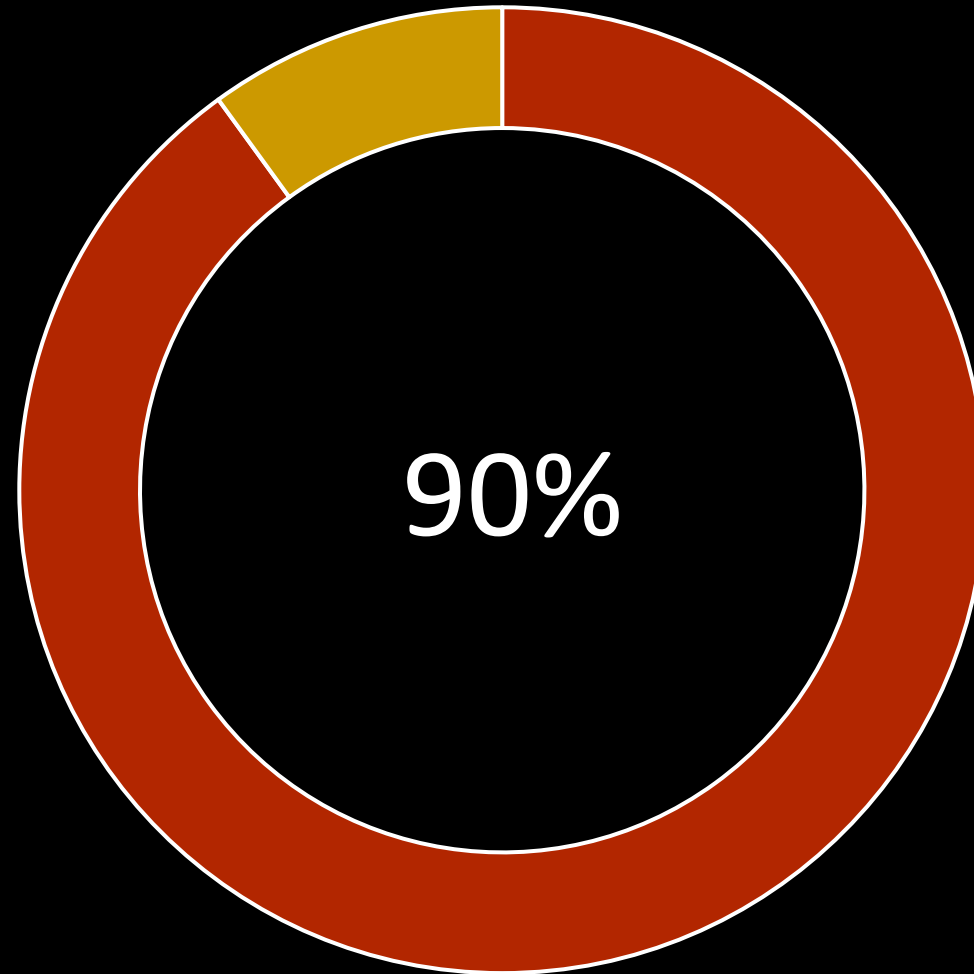


Open source is defining the new industry standards



Source: <https://github.com>. August, 2019.

90% of IT leaders are using enterprise open source.

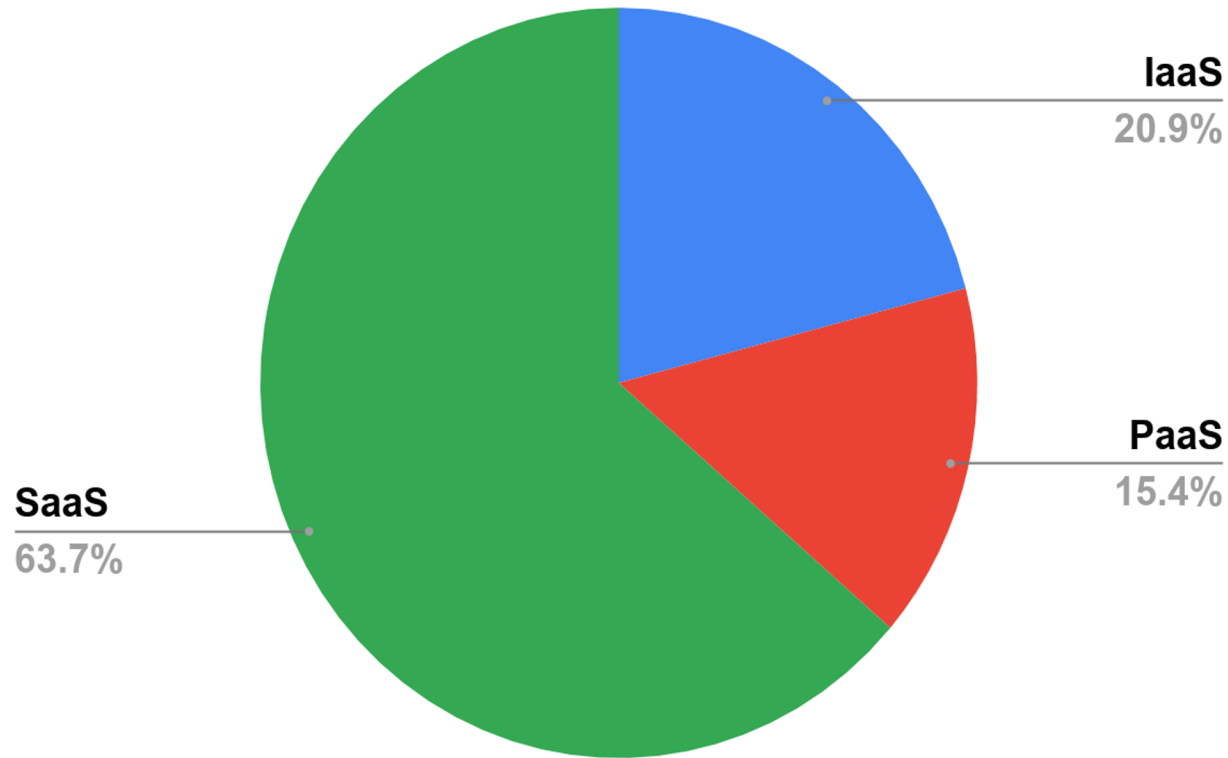


Source: Red Hat State of Open Source Report 2021

Cloud



Cloud Numbers



2020 Worldwide Public Cloud Revenues
~\$235B USD

Public Cloud

- IaaS - Dominated by 6-8 Clouds
- PaaS - Dominated by 50-100 Clouds
- SaaS - Over 4000 SaaS offerings

Data Center | Private Cloud

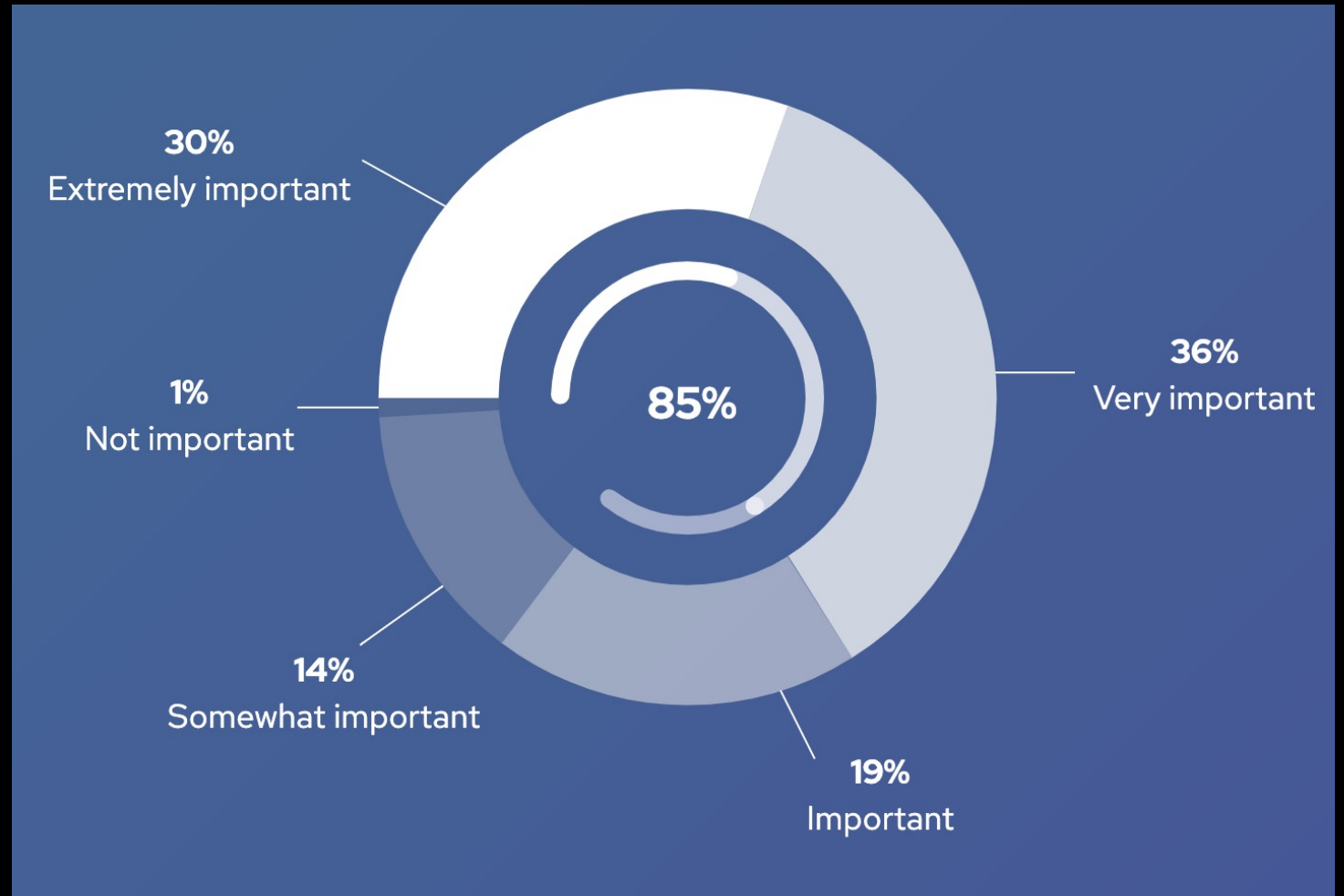
- "Only 20% is in the Public Cloud" - IBM
- "Only 5% is in the Public Cloud" - AWS

2020 Worldwide Data Center Revenues
\$2-4T USD

Kubernetes



85% of global IT
leaders agree
that Kubernetes is
key to cloud-native
application strategies



Source: Red Hat State of Open Source Report 2021



The Future

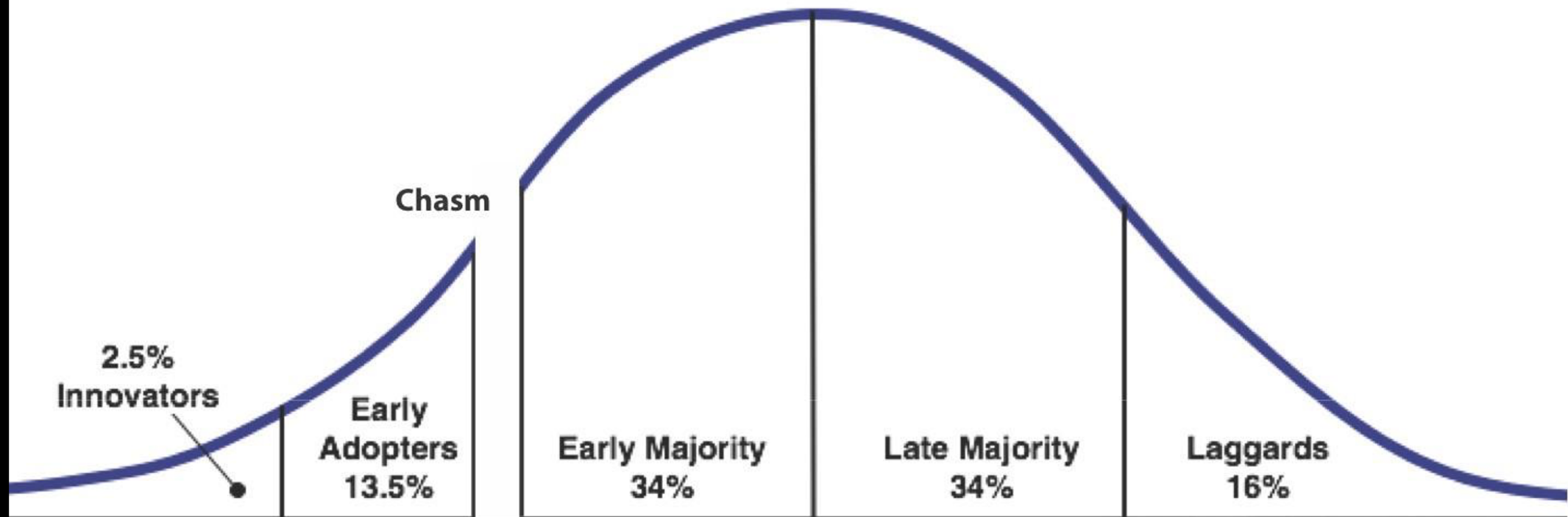


The future is already here.

It's just not evenly distributed

~ William Gibson

Technology Adoption Life Cycle



seeking advantage

seeking legitimacy

Agile

DevOps

SRE

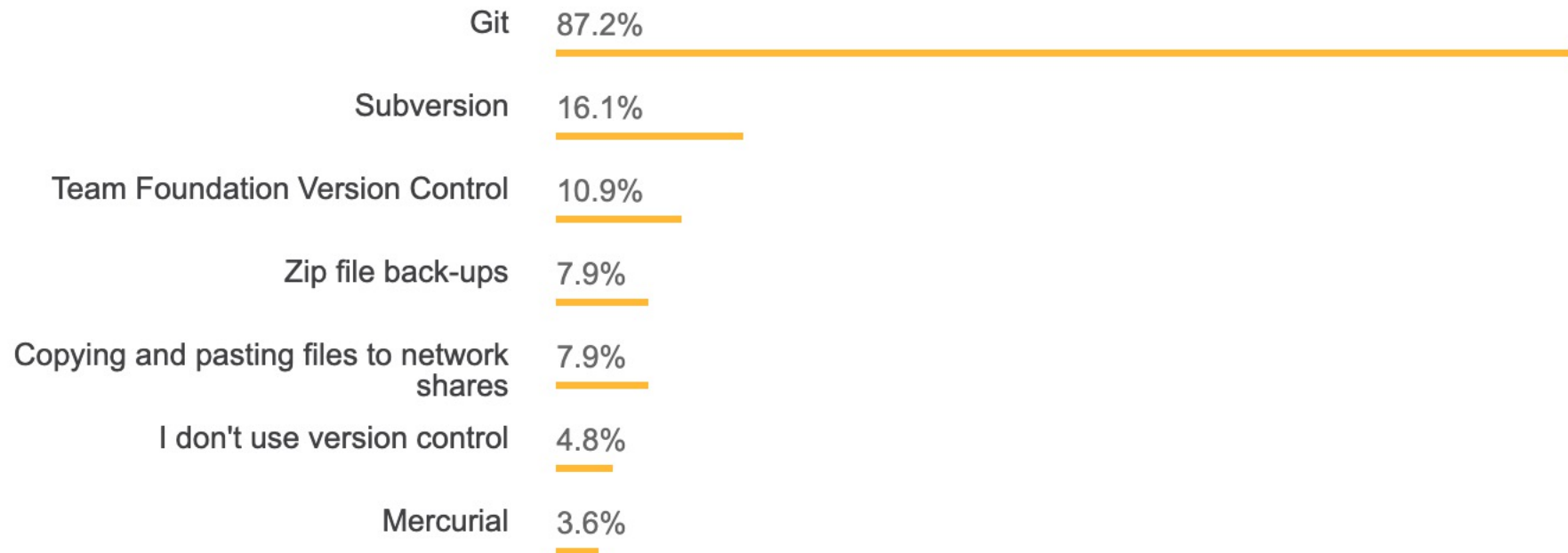
“DevOps is a solved problem”

- Someone from Google, 2019

Version Control

All Respondents

Professional Developers

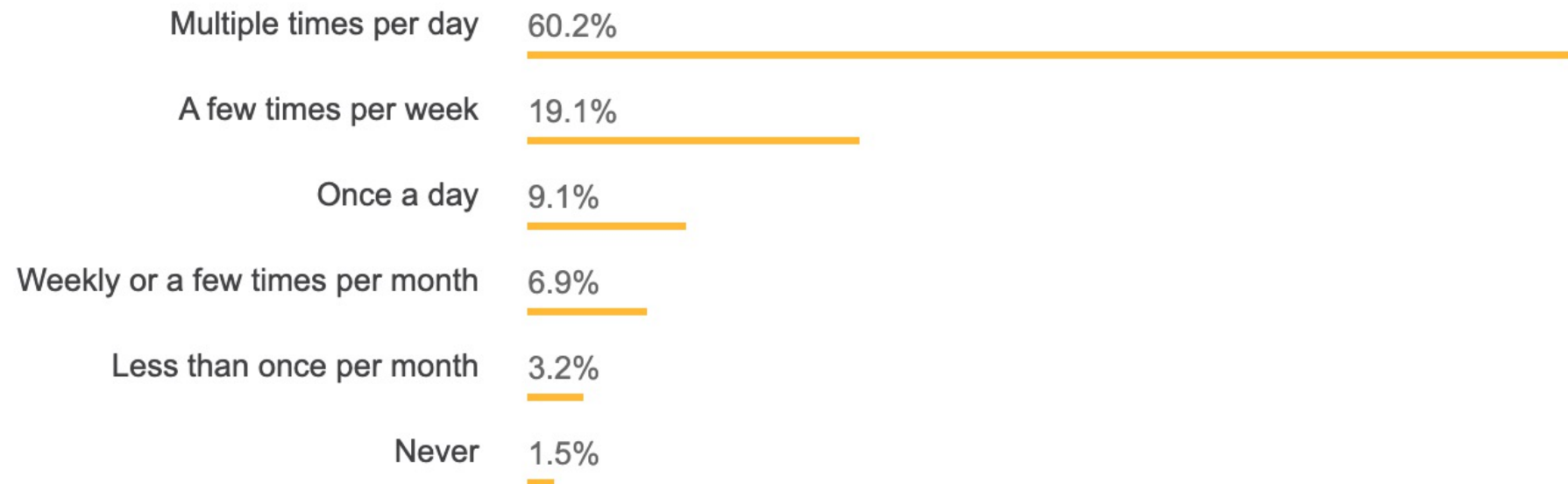


Source: StackOverflow Developer Survey, 2018

How Often Do Developers Check In Code?

All Respondents

Professional Developers



Source: StackOverflow Developer Survey, 2018

If knowledge
was all it
took,

we'd all have
six pack abs.



We must spend time on making
sure that the “standard of living”
improves for everyone

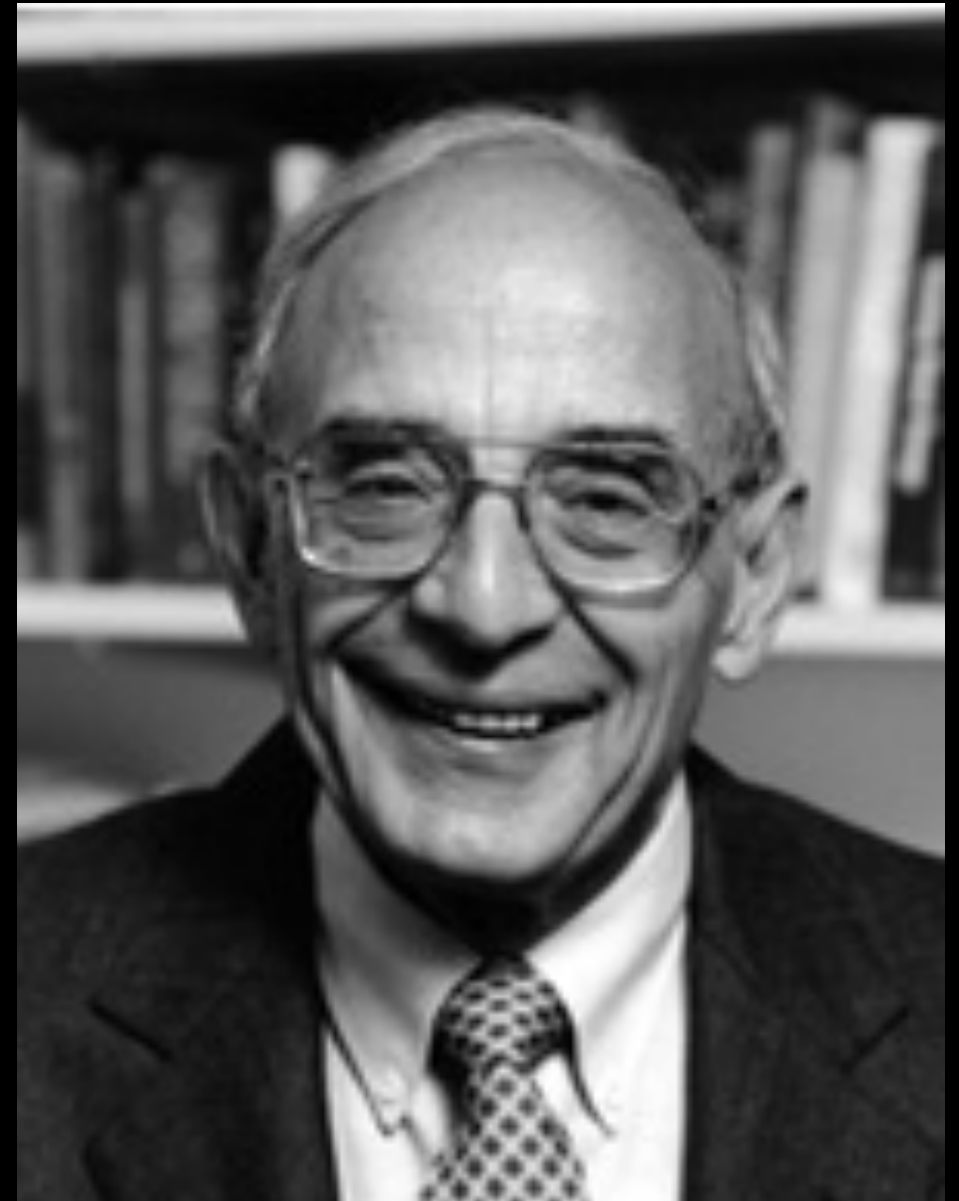
Companies,
just like people,
don't like to change

"Smart people don't learn ...

because they have too
much invested in proving
what they know

and avoiding being seen as
not knowing."

- Chris Argyris



Learning requires vulnerability

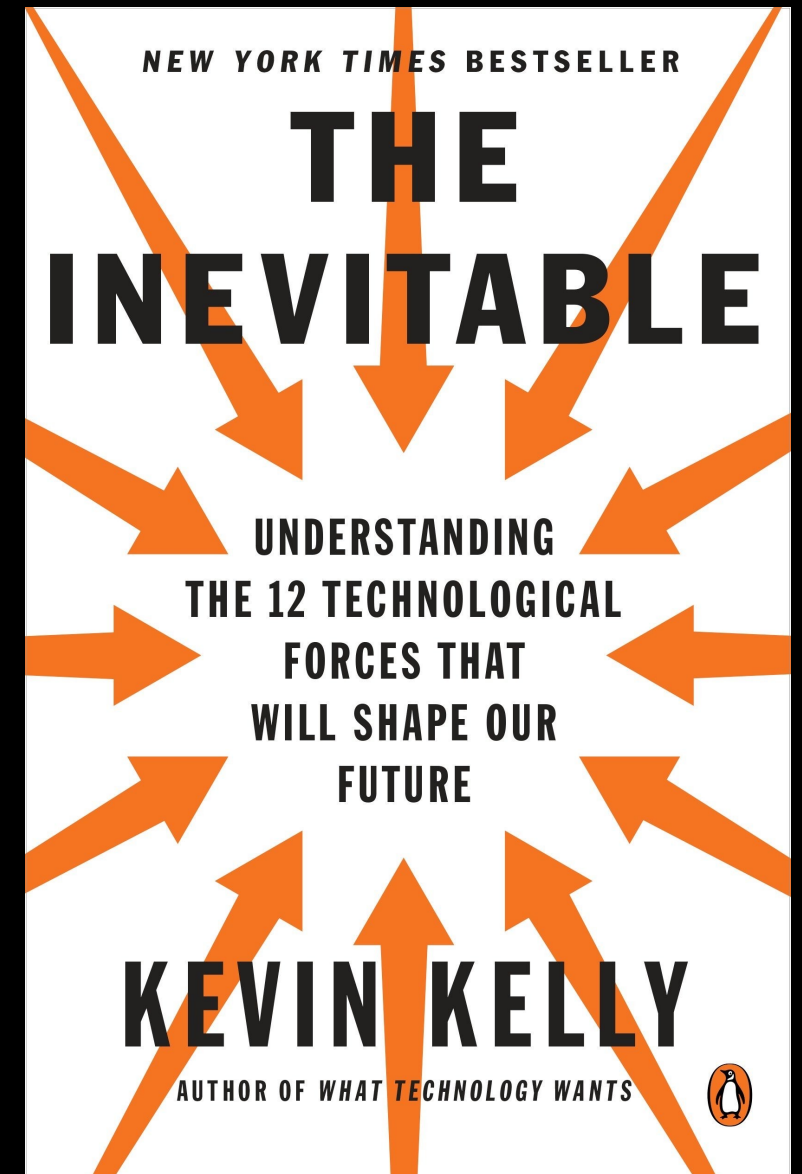
The age of continuous updates

Upgrading Windows



“In this era of becoming, everyone becomes a newbie. Worse, **we will be newbies forever.**”

- Kevin Kelly

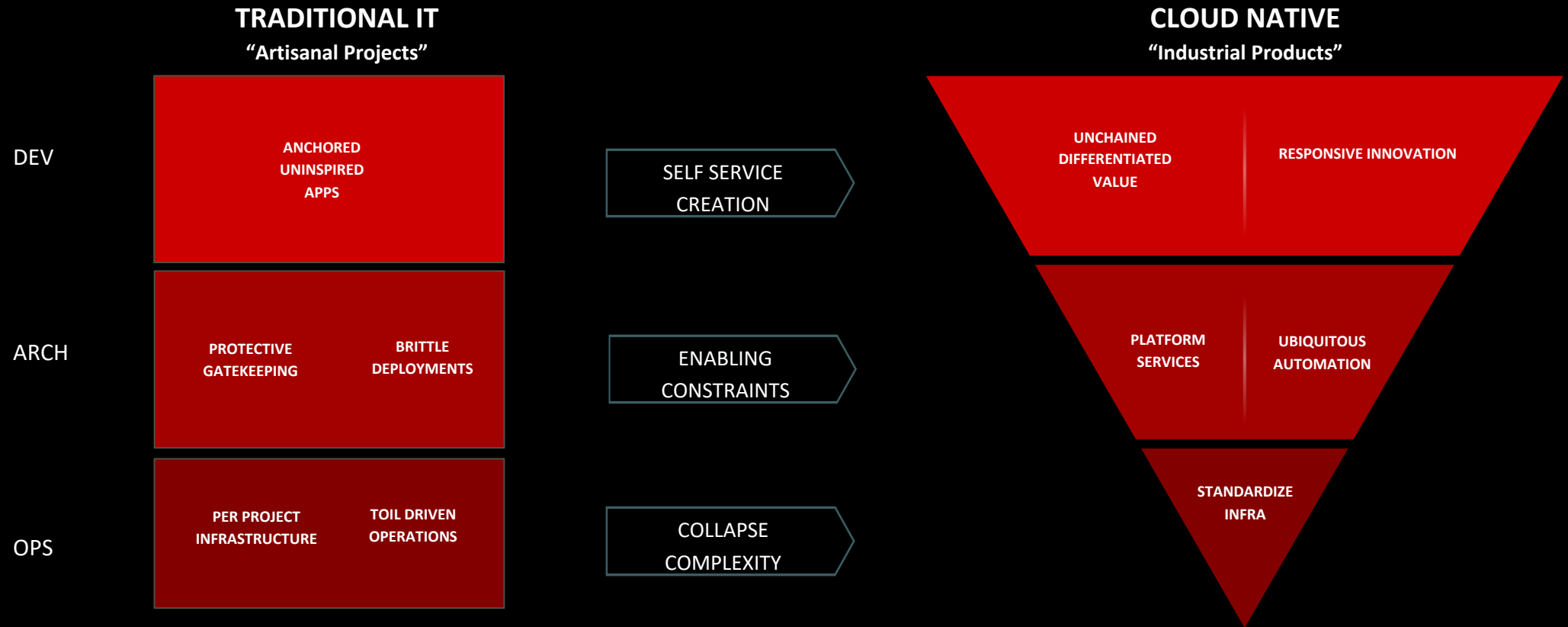


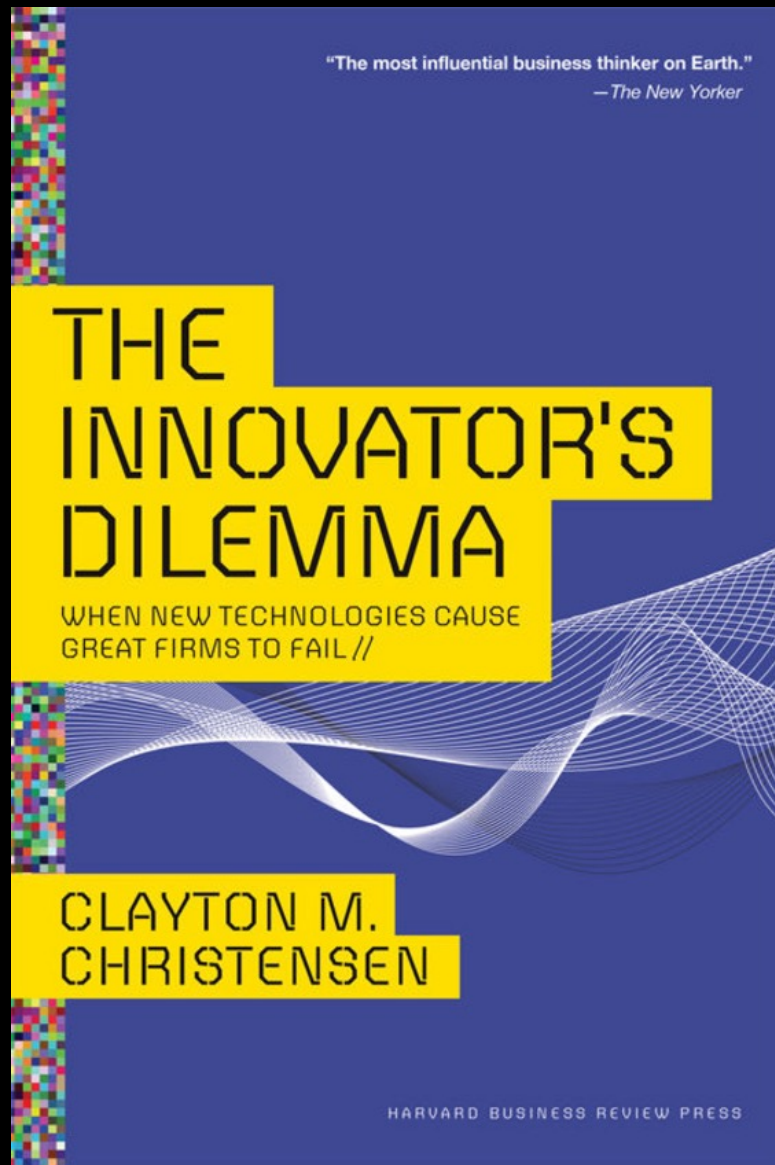
More innovation in the new
DevOps disciplines

Managed Services

Spend More Time On What Matters

The Cloud Native Organization





“This is one of the innovator’s dilemmas: Blindly following the maxim that good managers should keep close to their customers can sometimes be a fatal mistake.”

- Clayton Christensen

The DevOps evolution continues, as
we solve new problems every day

Good DevOps copy

Great DevOps steal



Thank you!

@DivineOps