

Future of DevOps

Sasha Rosenbaum
@DivineOps



Ergonautic



Sasha Rosenbaum

@DivineOps

Dev
Ops
Product
Sales
Consulting





The Past



1990s:

Getting a new
server up:
3+ months



@ergonauticly

COPYRIGHT © 2021

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.



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



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



GIT MERGE





Every company
ran its own email



Software as a Service

- Hotmail 1996
- Google Search 1998
- Salesforce 1999

Revenue share in 1999

- Application Service Providers \$933M
- Software Applications \$74B



Source: <https://www.inc.com/magazine/20000401/18093.html>

Deployment Checklists

Deployment Plan Template: Blue Theme

Deployment Plan Template - Blue Theme - Coverage 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	Stakeholders/Owners	7
1.5	Assumptions	7
1.6	Risks	7
1.7	Comments	8
1.8	Dependencies	8
2	Site Information	9
2.1	Site Overview	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 Administrator	12
3.3	Installation Manager	12
3.4	Clear Technology Operator	13
3.5	Hardware Installation Specialist	13
3.6	Software Support Engineer	15
3.7	On-demand Subject Matter Experts (SMEs)	15
3.8	Project Sponsor	15
4	Schedule	17
4.1	Resource Allocation Chart	17
5	Resources	19
5.1	Facilities	19
5.2	Hardware	19
5.3	Software	19

54 Documentation

61	Roll-Over	21
62	Overview	21
63	Summary	21
64	Revision History / Change Management	21
7	Testing	22
7.1	Tested	22
8	Installation Considerations	24
8.1	Survey	24
8.2	Clear Installation Requirements	24
8.3	Installation Planning	24
8.4	Review & Readiness	25
8.5	Installation Site Maintenance	25
8.6	Installation Records	25
9	Installation Process	26
9.1	Preparation	26
9.2	Installation	27
9.3	Testing	29
9.4	Site Maintenance	29
10	Transfer to Operations	31
10.1	Preparation	31
10.2	Preparation	31
10.3	Equipment Arrival or Change	31
10.4	Preparation Activities	31
10.5	Readiness	31
10.6	Roll-Over	31
10.7	Reception or Decommission	31
11	Appendix	32
11.1	Attachments and Appendices	32

1 Introduction

The Deployment Plan is a "how-to" guide to implement a solution into production. In other words, to deploy on a target, install and activate, or decommission an existing environment. It is a document that provides the deployment guidelines and tasks that the deployment manager, in general, the system administrator, architect, helper, architect and hardware engineer will use to implement the deployment plan by performing the high-level activities necessary to get the target system into production or decommission. Depending on the scope of the deployment, this document will describe the installation, configuration, verification and testing activities that will be performed as part of the deployment process. As part of the deployment process, the following information may be relevant to your project: please refer to the template to modify your needs.

1.1 Purpose

Identify the purpose of the Deployment Plan and its intended audience, which is a subset or a combination of client management and members of the system integrator's team. Clarify the role.

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, testing, validating, and transferring the solution to daily operations as well as details on different installation scenarios, modeling for scalability, and ensuring 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.
- The installation includes activities involving the Client to facilitate the installation. Plan, Training Plan, Communication Plan, Schedule and other deployment documents.
- Ensure that all safety precautions 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 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, subsystems, and software installations.

2 Site Information

Place the deployment site in context by describing how the target system fits within the deployment. Describe and reference to the specific specifications and other relevant documentation that will be referenced in the section.

Ref #	Issue	Impact
#1	Describe the issue	Discuss the effect on the deployment plan
#2	Describe the issue	Discuss the effect on the deployment plan
#3	Describe the issue	Discuss the effect on the deployment plan



2000



27% of server market

- File-based OS
- Maintains configuration in files
- Every device is a file

41% of server market

- Executable-based OS
- Maintains configuration in registry
- Every device has a different driver mechanism

1990s:

Maintenance windows



System Downtime

Saturday, June 4 -
Sunday, June 5



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.

Ergonautic 

How many 9s is two weekends like that?



< 99%

**3.65 days /
year**

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

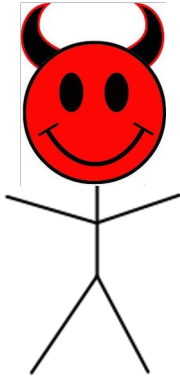


@ergonauticly

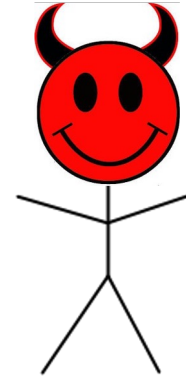
COPYRIGHT 2023. ALL RIGHTS RESERVED.



Traditional IT



dev



ops

wall of confusion

Dev

Ops



Speed

Reliability



© 2018 Forrest Brazeal



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



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



Andrew Clay Shafer 雷启理

@littleidea

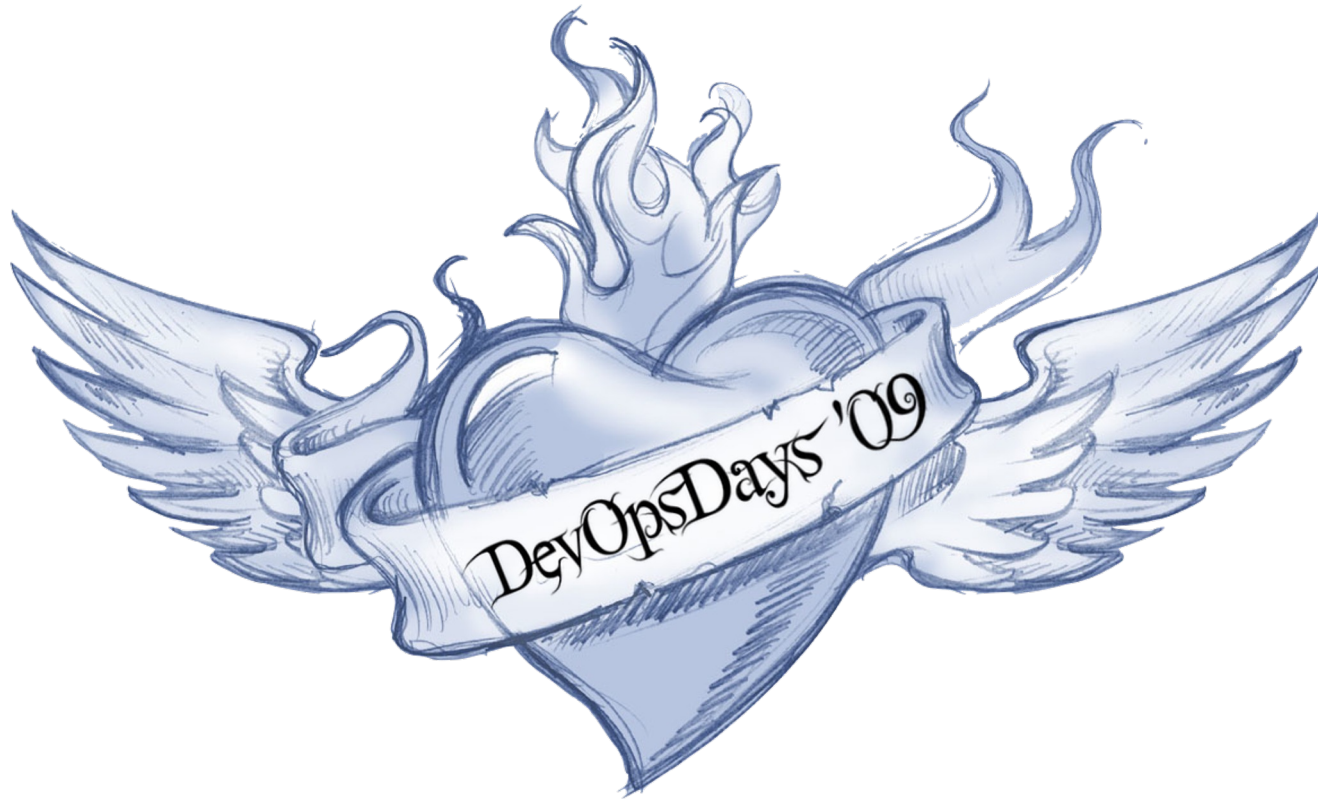
Don't just say 'no', you aren't respecting other people's problems... [#velocityconf](#)
[#devops](#) [#workingtogether](#)



@ergonauticly

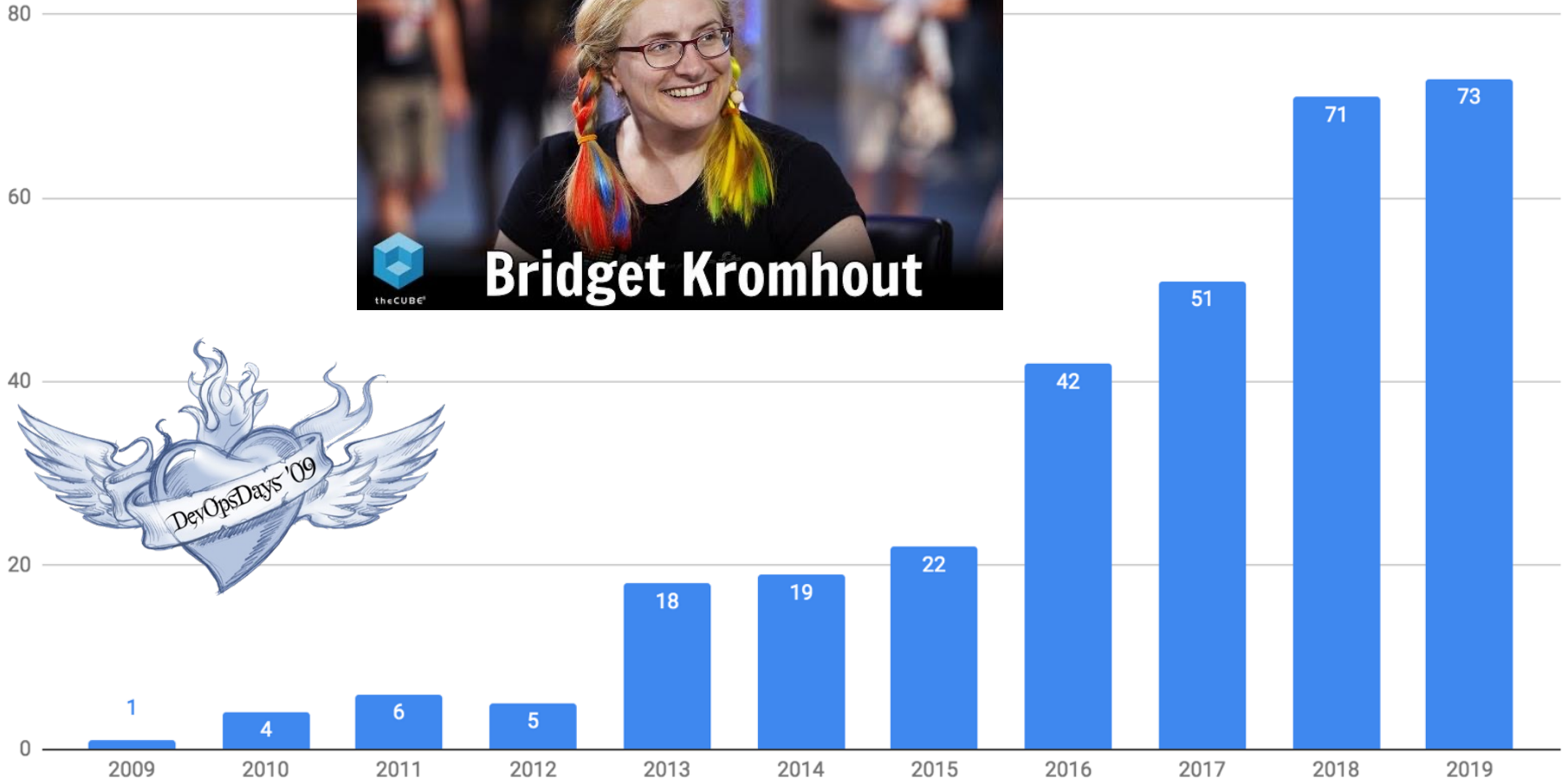
COPYRIGHT 2023. ALL RIGHTS RESERVED.

www.ergonautic.ly



DevOpsDays Ghent 2009: Patrick Debois

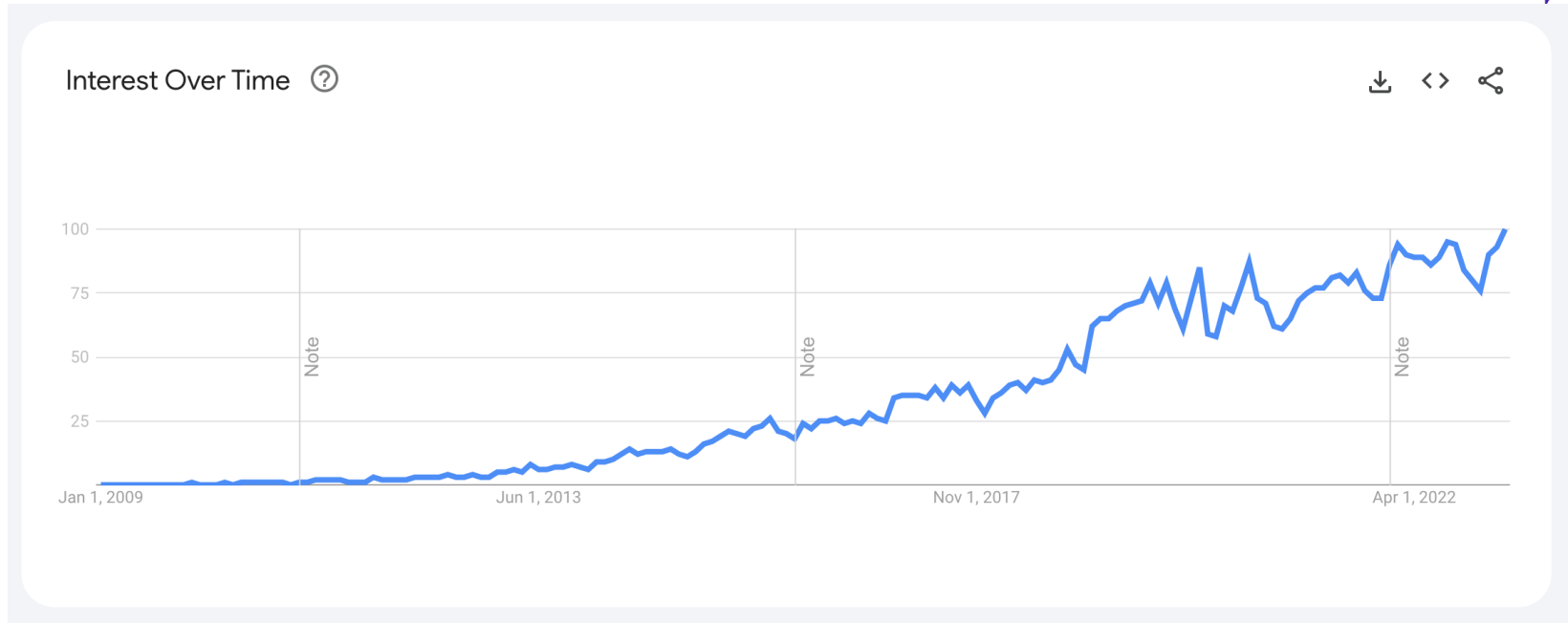
devopsdays events



“DevOps” Google Search Trend 2009-Present



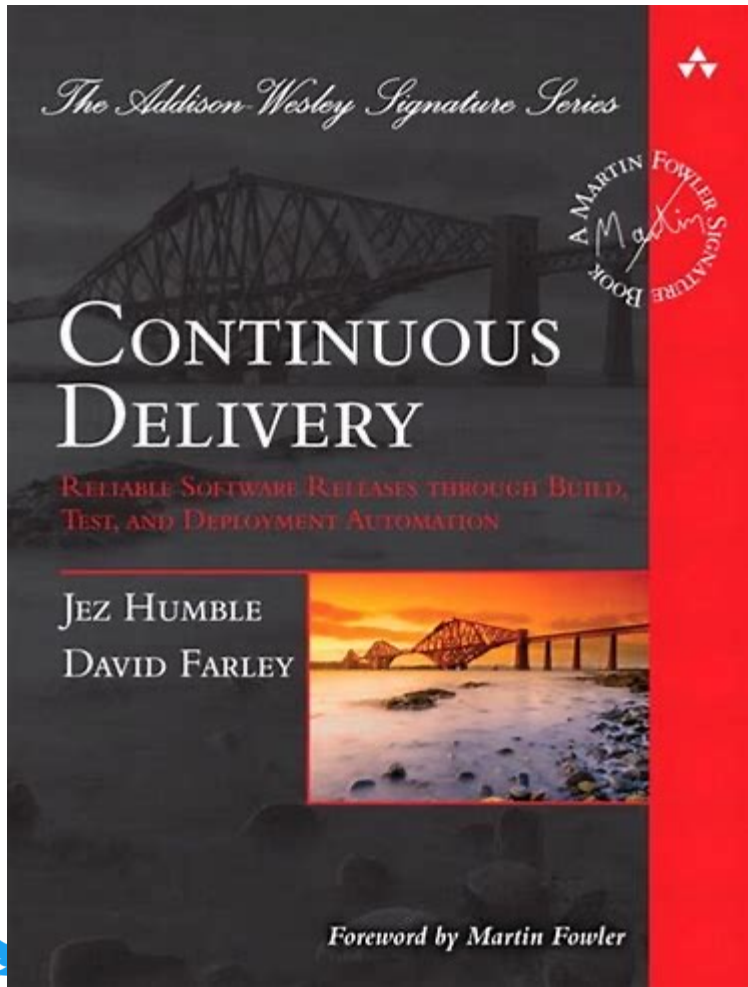
Ergonautic



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.

www.ergonautic.ly



Jez Humble and Dave Farley: 2010

Continuous Integration (CI)



The practice of **merging** code into the main branch several times a day; and automating the **build and testing** on every **commit**.



Continuous Delivery (CD)



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



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 1/7 as likely to fail)



Throughput Stability



Nicole Forsgren. State of DevOps Report 2019



Software delivery is like a muscle.

The more you use it, the stronger it gets.



The Present

 @ergonauticly

COPYRIGHT 2022. ALL RIGHTS RESERVED.



Ergonautic 

Effective automation
requires consistent APIs



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



OS-level APIs

PowerShell

(Windows) configuration management framework and scripting language



Jeffrey Snover, 2006

Infrastructure-level APIs

Amazon Web Services: 2002

Amazon Cloud Computing: 2006

Azure Cloud Services: 2008



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Infrastructure as code

2005



2009



2012



ANSIBLE

A large, powerful blue wave crashing, symbolizing automation. The wave is curling over, creating a tunnel-like structure. The water is a deep blue, and the foam is white. The background is dark, making the wave stand out.

Every wave of
automation

Enables the next
wave of automation

Today



 @ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.

www.ergonautic.ly



We have a lot more automation



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



We have a lot more automation tools



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



We have much higher availability

We have better on-call

We have better **incident response**



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



We deploy a lot more frequently

BUT



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Many of us are bad at monitoring



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Many of us are bad at CI

Continuous Integration (CI)



The practice of **merging** code into the main branch several times a day; and automating the **build and testing** on every **commit**.



You cannot CI/CD Without Testing

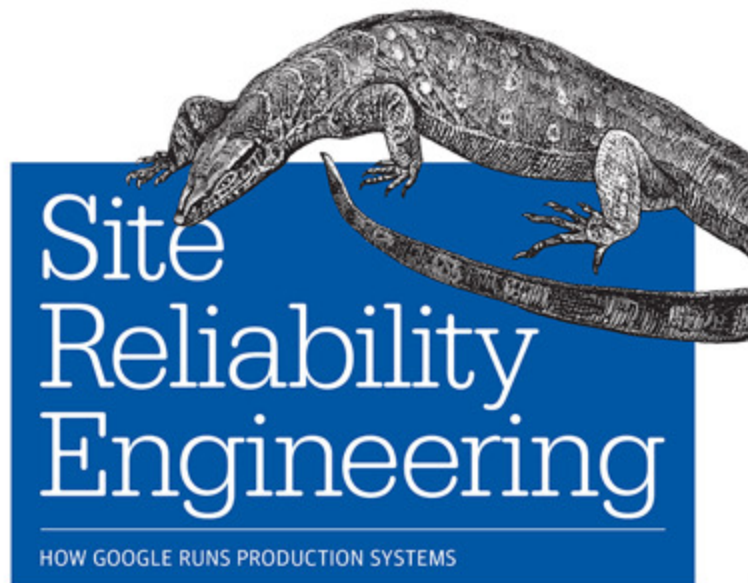


We have convinced people that
deploying faster is good

We forgot to mention that increasing
the Operational Burden is bad



O'REILLY®



@ergonauticly

Edited by Betsy Beyer, Chris Jones,
Jennifer Petoff & Niall Murphy



SRE \approx Google's implementation DevOps



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Why did SRE develop?

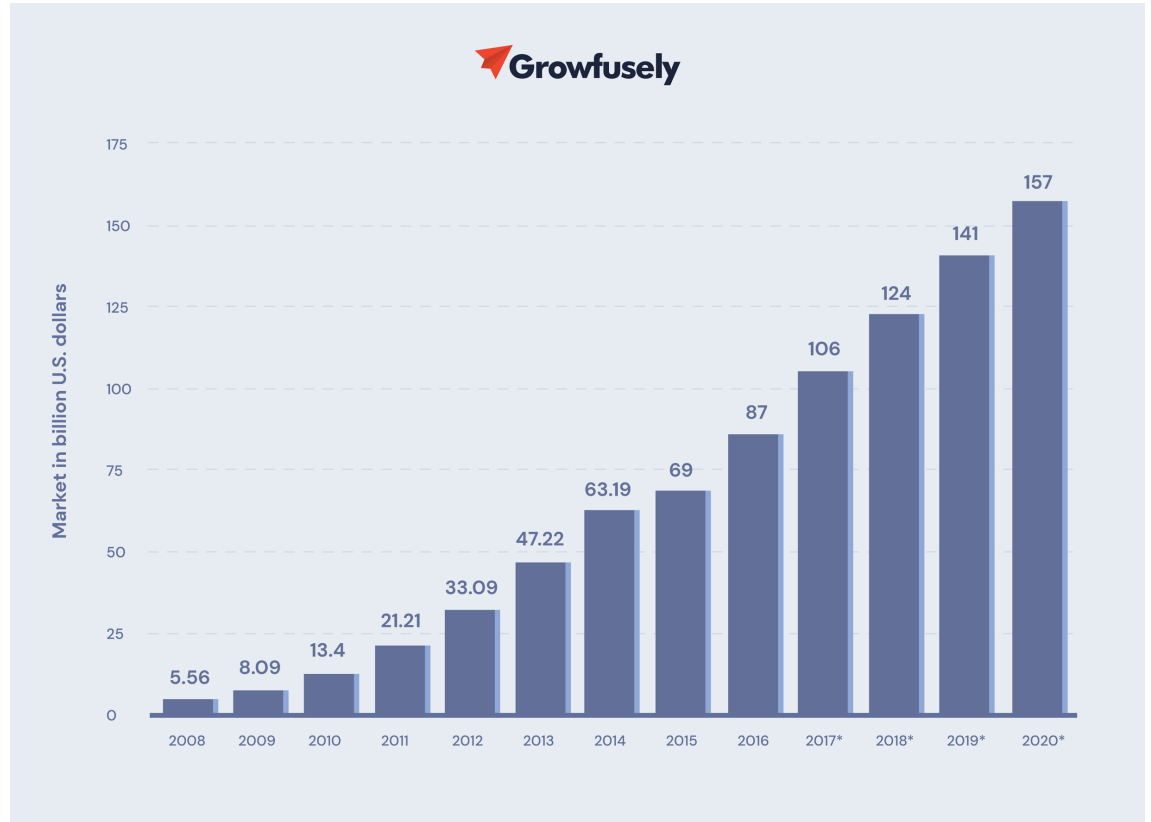


@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



SaaS Market 2008-2020



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



What is the most important thing
about the SRE discipline?



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



SRE is about explicit agreements
that align incentives



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



SLA, SLI, SLO

SLA

=

Financially-backed availability



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Monthly Uptime Percentage

Service Credit Percentage

Less than 99.95% but equal to or greater than 99.0%

10%

Less than 99.0% but equal to or greater than 95.0%

25%

Less than 95.0%

100%

Monthly downtime > 15 days means 100% refund

SLAs are about
aligning incentives
between Vendor & Customer

SLO

=

Targeted reliability



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



While SLA is a **single** metric,
SLOs are a **system** of metrics

SLO

=

Business-approved reliability



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



SLOs are about
explicitly aligning incentives
between Business &
Engineering



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.





Error Budgets

Acceptable level of **un**reliability

Error budget = 1 - SLO

$EB = 1 - 99.99\% = 0.01\% \simeq 13 \text{ mins /quarter}$



Error budgets are about
aligning incentives
between Dev & Ops



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



If **developers** are measured on the same **SLO**,
then when the **error budget** is drained
developers shift focus from delivering new
features to **improving Reliability**



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



SL1

=

Actual reliability



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Monitoring



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Without **monitoring**, you have no way to
tell whether your service even works!



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Good Monitoring



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Without **good monitoring**, you don't know that the service does what users expect it to do!

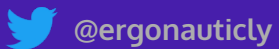


@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



The Future



COPYRIGHT 2022. ALL RIGHTS RESERVED.



Something we have to talk about



AI



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.

Ergonautic 

AI will not [yet] take your job



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



AI will change jobs



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Ethics



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Like a Library



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Current **AI** aggregates and synthesizes the results of decades of human work



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Written works, code, art



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Unlike a Library



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.





Current **AI** charges its users for
the content

But it **DOES NOT** pay royalties or
give attribution to the creators



Fairness usually loses to profit



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Are people going to stop sharing?



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Something else we have to talk about

DevOps is Dead



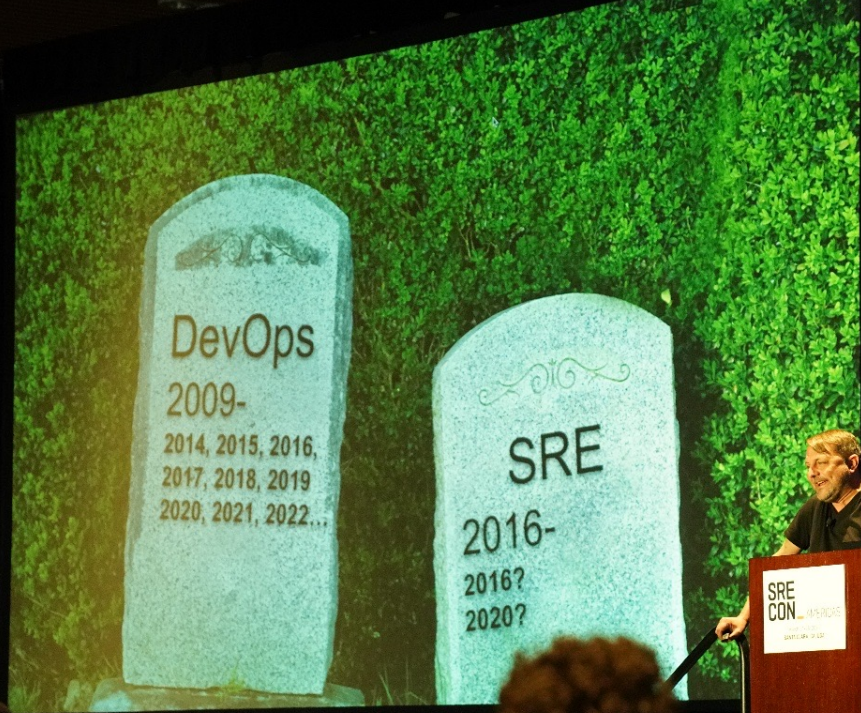
@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Is it though?





DevOps

2009-

2014, 2015, 2016,
2017, 2018, 2019
2020, 2021, 2022...

SRE

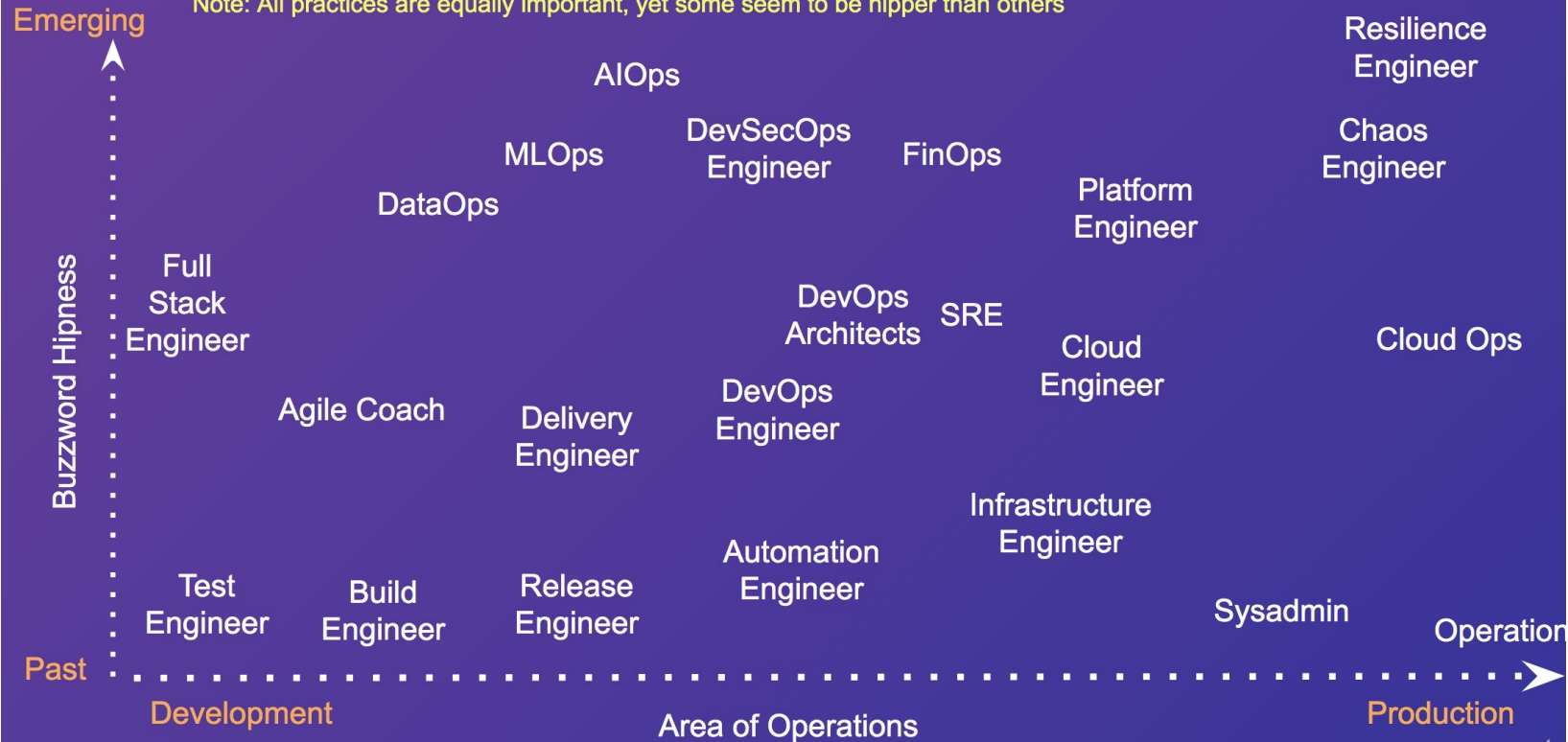
2016-

2016?
2020?

SRE
CON

Shades of DevOps Industry Terms

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



Patrick Debois, 2021

Someone in your org is keeping
your systems alive

And their life might be hell

The future is already here.
It's just not evenly distributed

~ William Gibson



@ergonauticly

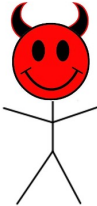
COPYRIGHT 2023. ALL RIGHTS RESERVED.



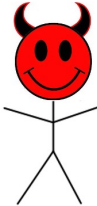
Renaming Teams



Support -> Sys Admin -> Ops -> DevOps -> SRE -> Platform Eng (?)



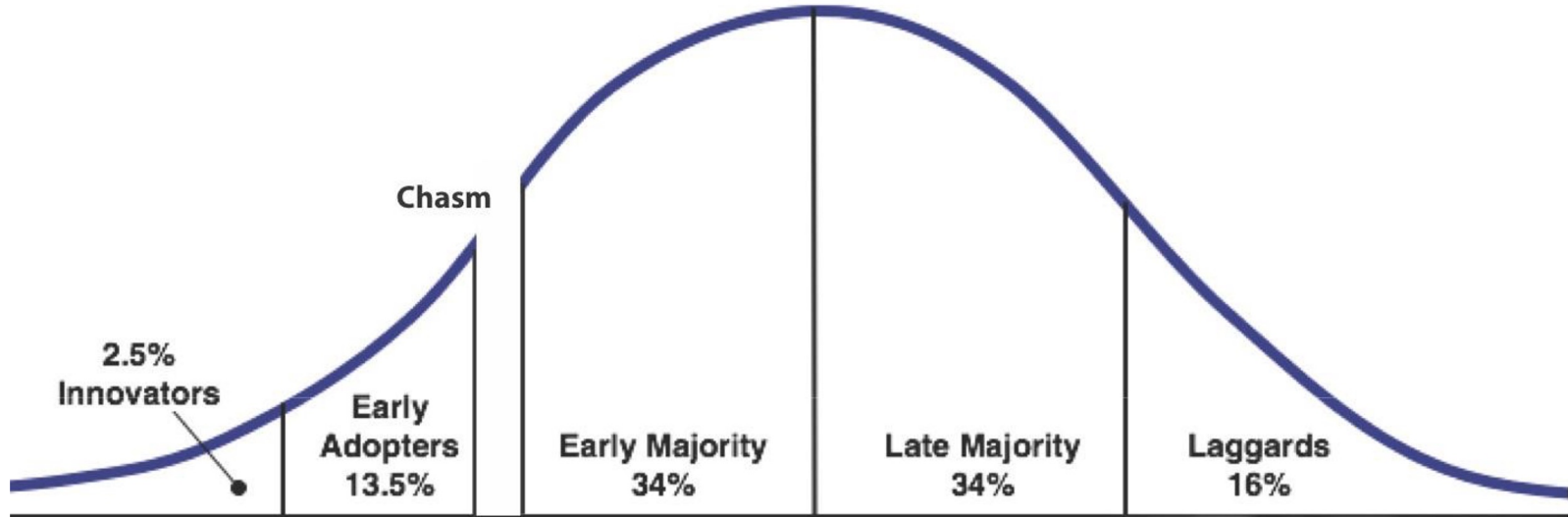
dev



ops

wall of confusion

Technology Adoption Life Cycle



seeking advantage

seeking legitimacy

words cross the chasm before
understanding and practice

Most organizations I encounter,
even those running SaaS,
have no SRE practice



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Most Organizations today



- Don't **merge** code frequently
- Don't have quality **testing** during CI
- Don't have informative **monitoring**
- Don't enforce **Error Budgets**
- Don't have a **platform**



everyone wants DevOps

well actually...

what they really want

- reliability
- availability
- scalability
- operability
- usability
- observability
- all for free
- without changing anything

without
changing anything

without
changing anything

without
changing
anything

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Software

← Everyone wants this

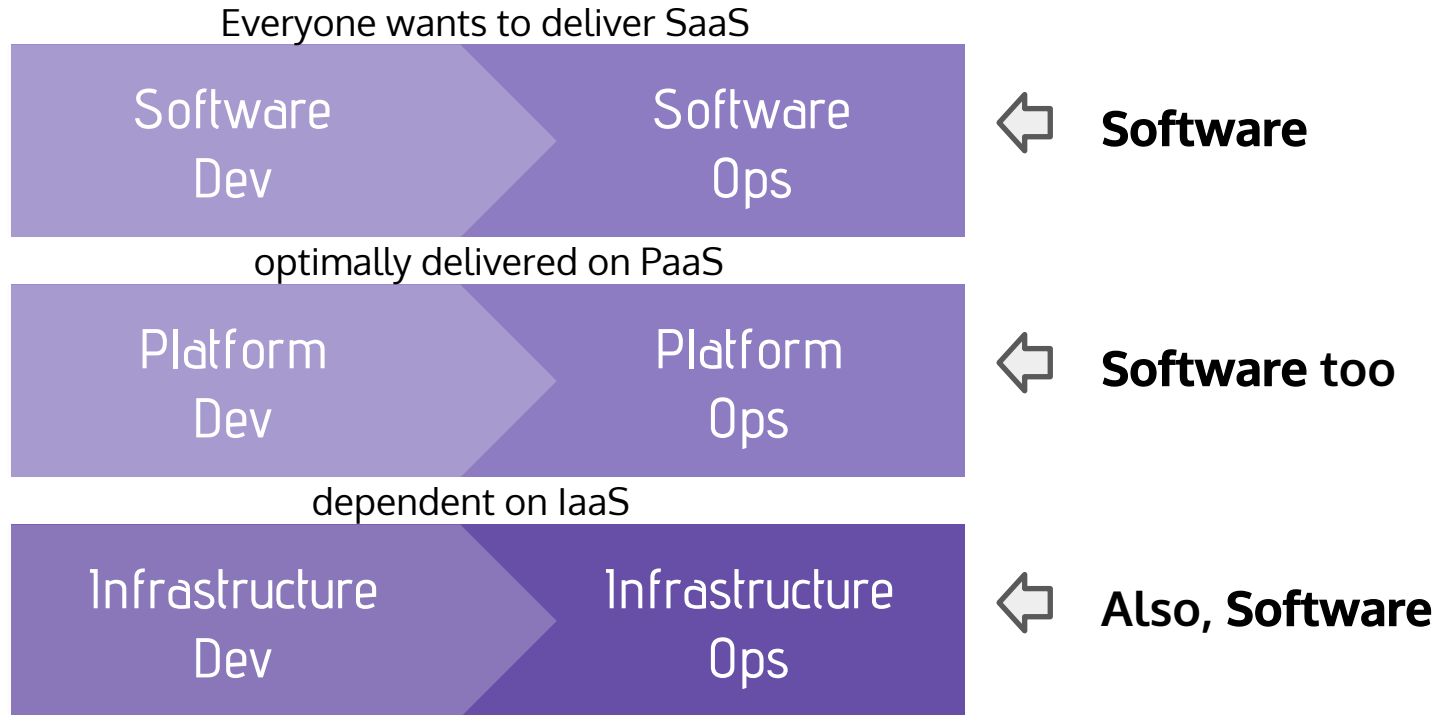
Platform

← When they haven't solved this

Infrastructure

← Or even this

Operating Model



Let's bring the future to everyone

We don't need new words

We need to implement what we
already know to work well

Let's start small



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Not this



This



The future is already here.
It's just not evenly distributed

~ William Gibson



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.



Thank you!

Sasha Rosenbaum

@DivineOps



@ergonauticly

COPYRIGHT 2023. ALL RIGHTS RESERVED.

www.ergonautic.ly

