2020-04-01
Remote Tech Night

Site Reliability Engineering

Sven Johann

INNOQ

Sven Johann

Senior Consultant
bei INNOQ Deutschland GmbH

Run the systems I develop for 10+ years
Community guy (GOTOcon, TechDebtConf, CaSE Podcast)

What is Reliability?

Software Architecture (ISO 25010)

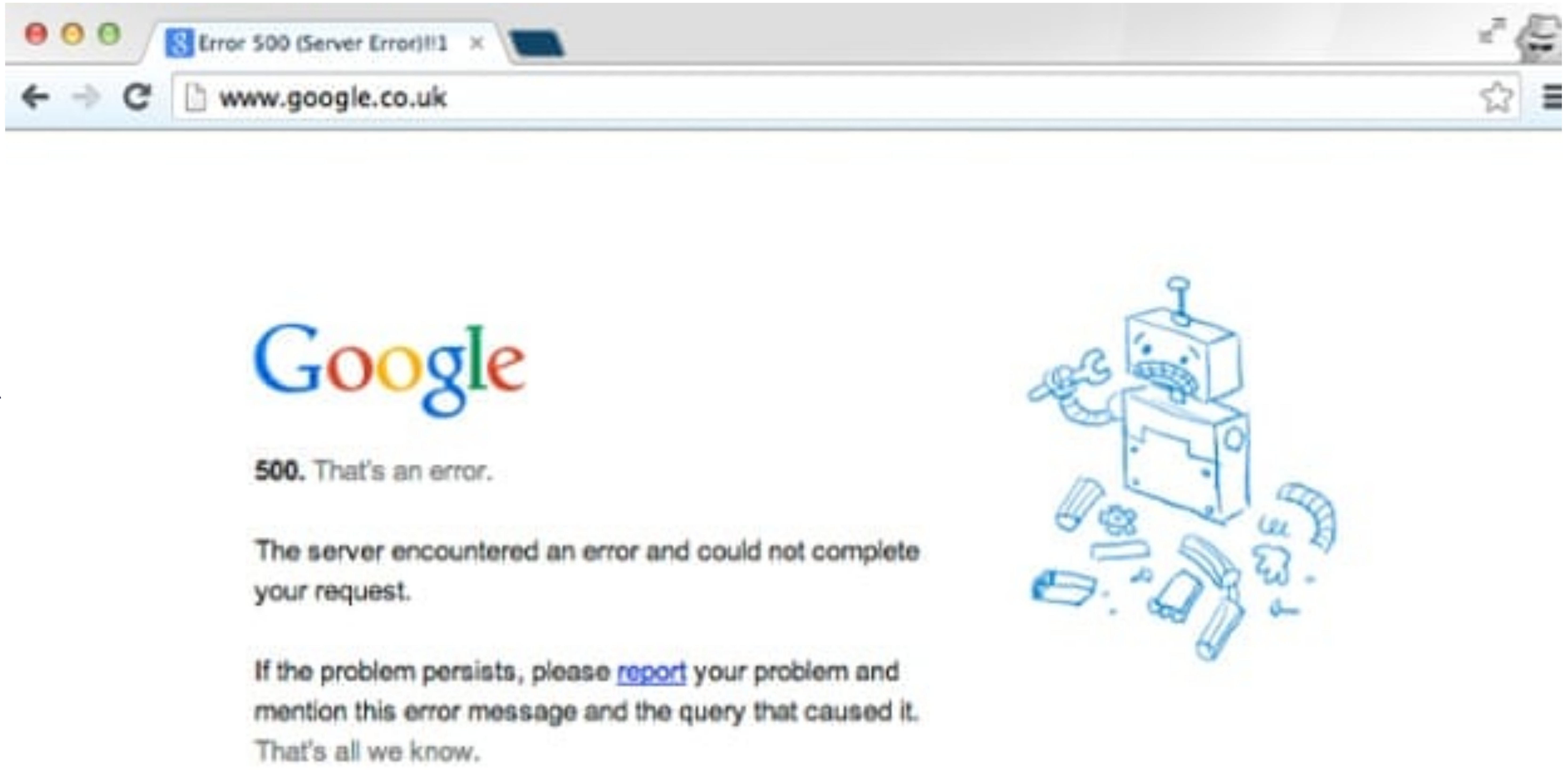
- availability, fault tolerance, recoverability, maturity

Google SRE responsibilities

- availability, latency, performance, efficiency, change management, monitoring, emergency response, and capacity planning

Why reliability?

If your application cannot be used, what are your nice features worth?


Error 500 (Server Error)!!1

www.google.co.uk

500. That's an error.

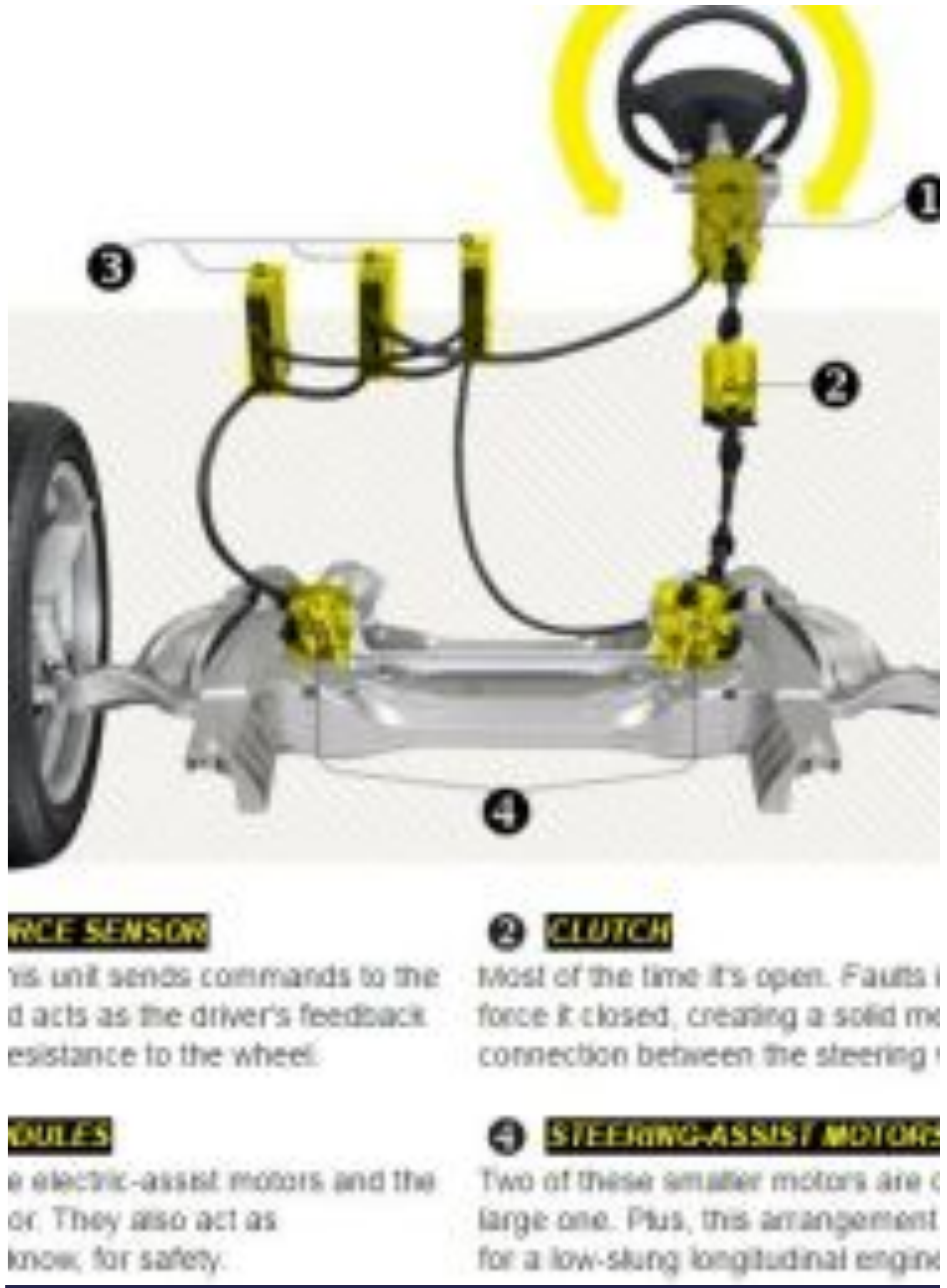
The server encountered an error and could not complete your request.

If the problem persists, please report your problem and mention this error message and the query that caused it. That's all we know.

Source: https://www.theguardian.com/technology/2014/jul/04/google-down-search-services-intermittent-outage

How much reliability?

Pace makers, x-by-wire in cars and plains need extremely high reliability



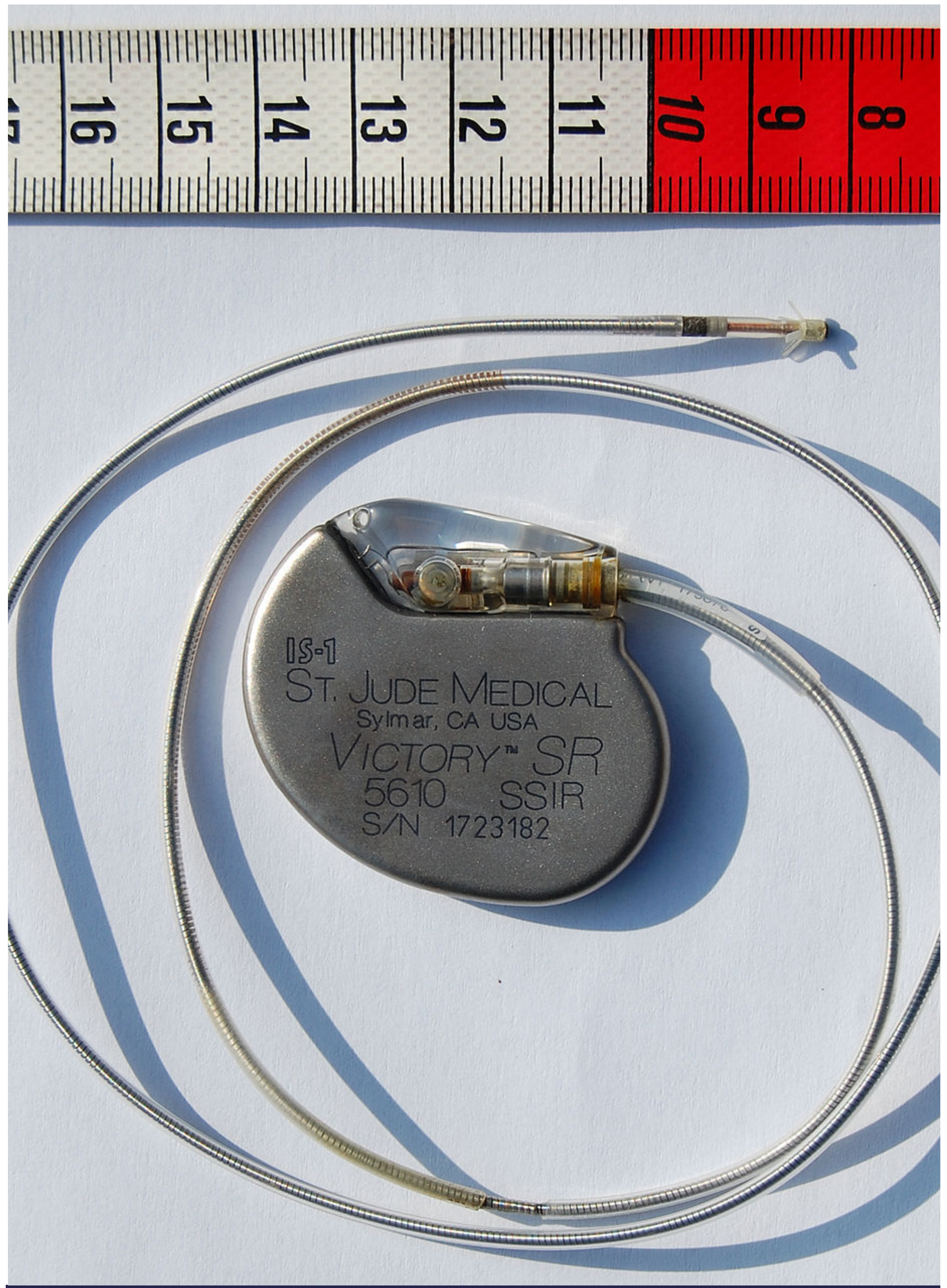
RCE SENSOR
his unit sends commands to the d acts as the driver's feedback esistance to the wheel.

2 CLUTCH
Most of the time it's open. Faults force it closed, creating a solid me connection between the steering

DULES
e electric-assist motors and the or. They also act as know, for safety.

4 STEERING-ASSIST MOTORS
Two of these smaller motors are large one. Plus, this arrangement for a low-slung longitudinal engine

Steer-by-wire: https://mymotorwheels.wordpress.com/2017/02/10/have-you-ever-wondered-what-is-drive-by-



https://en.wikipedia.org/wiki/Artificial_cardiac_pacemaker

How much reliability?

Google Ads makes 4000 USD per second

<u>amazon.com</u> **retail makes 5000 USD per second**

How much reliability?

Finance: usually "medium"

SaaS usually "high"

Retail usually "high"

Websites not generating much revenue usually "low"

Platforms and developer

ING

salesforce

OTTO

Universität zu Köln

Amazon EKS

Reliability is often not well understood

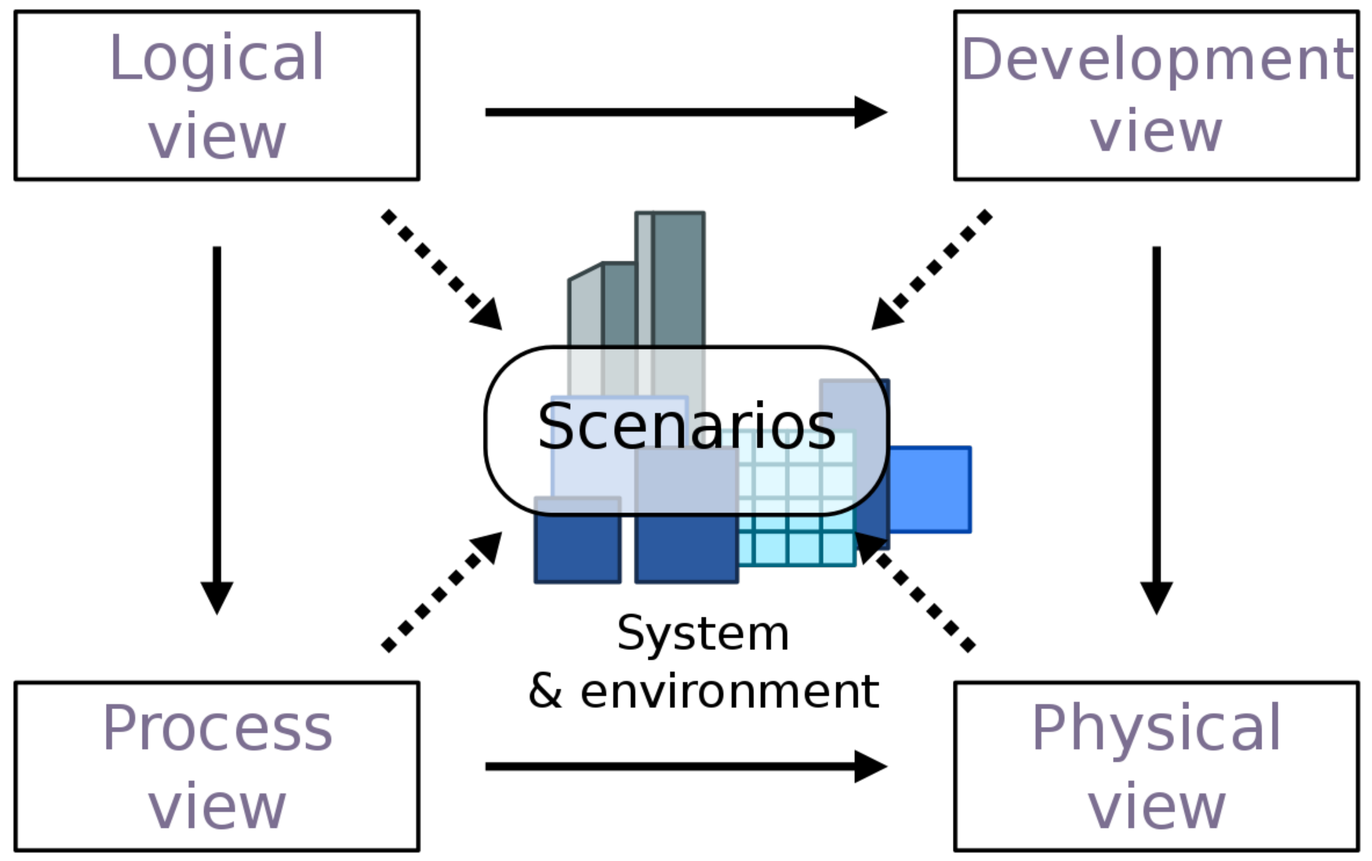
- People expect systems be available 100% of the time or "as much as possible"
- Availability comes with a cost. You need to make cost/benefit trade-offs
- Invisible: the absence of errors
- If your system is unreliable, it is already too late. Fix is often hard
- It is continuous work and not fire fighting

Causes for Reliability Problems?

- "You build it, you run it" often suffers from inexperienced devs
- Operations is not treated as it should by lead developers and architects

The Pragmatic Programmers
Release It!
Second Edition
Design and Deploy Production-Ready Software
Michael T. Nygard
Edited by Katharine Dvorak

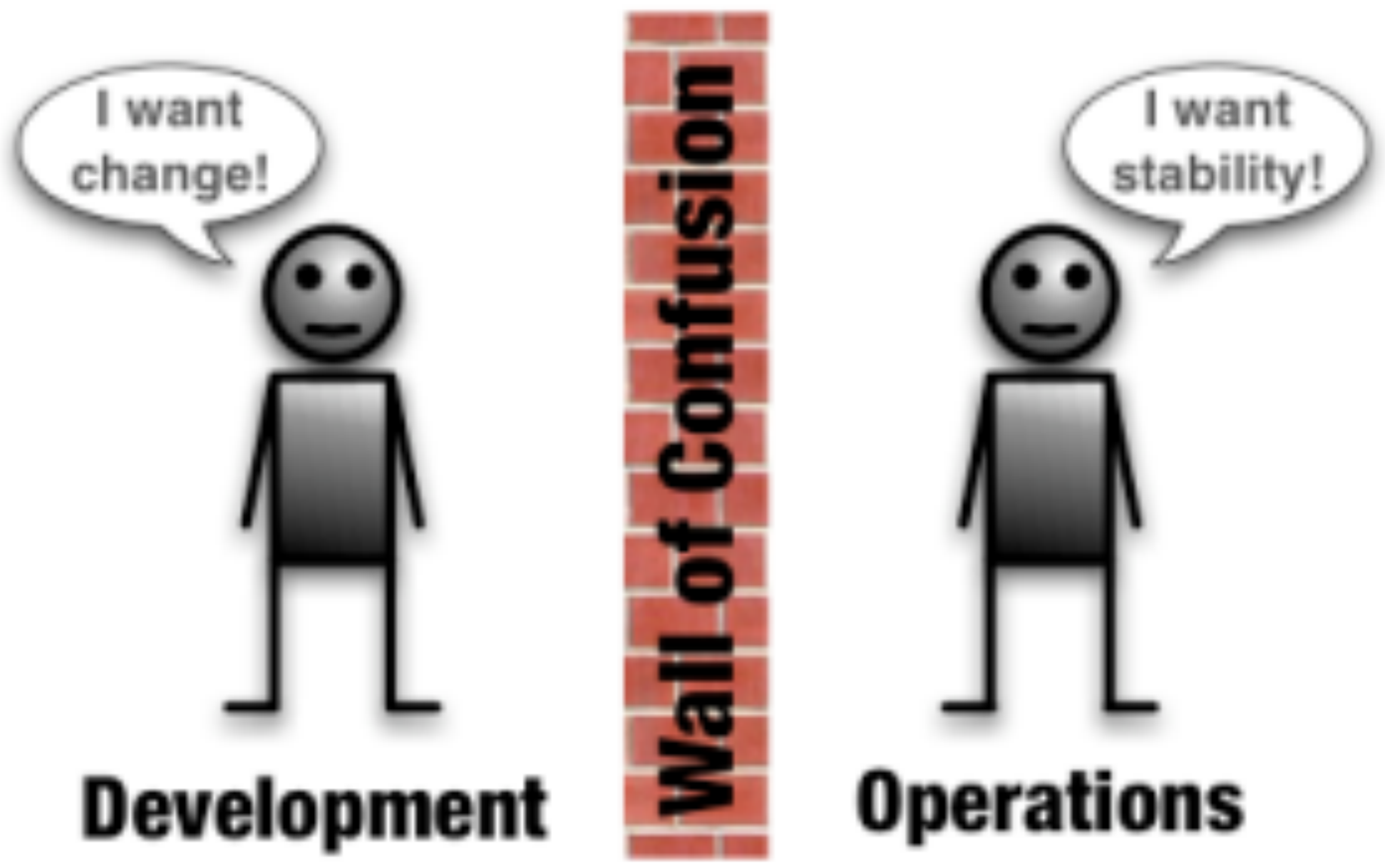
Release It, M. Nygard



Source: 4+1 Model, Wikipedia

Causes for Reliability Problems?

- Dev and Ops have conflicting goals
- Ops has no idea of the code they are running



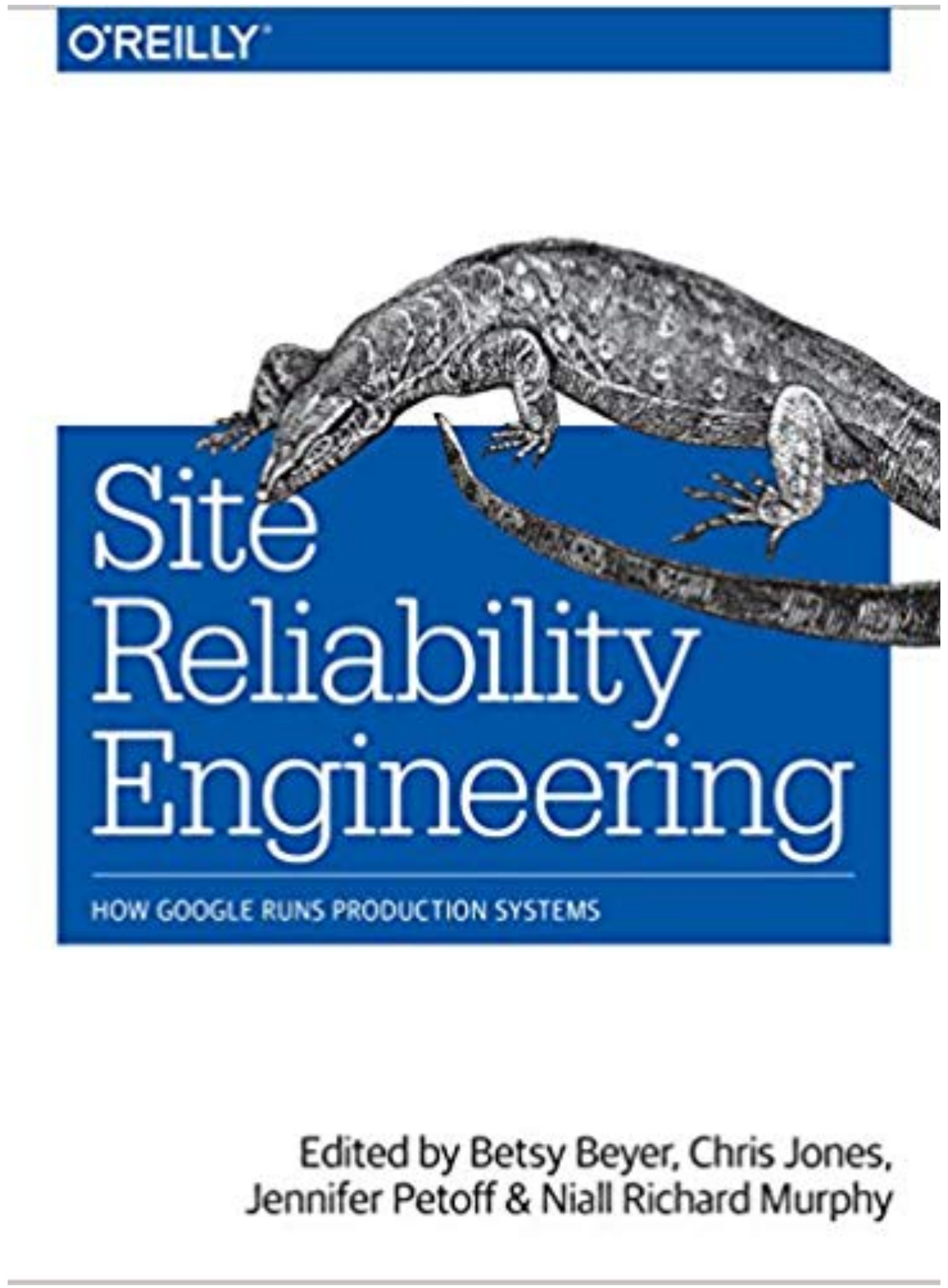
Source: Andrew Clay Shafer, Agile Infrastructure

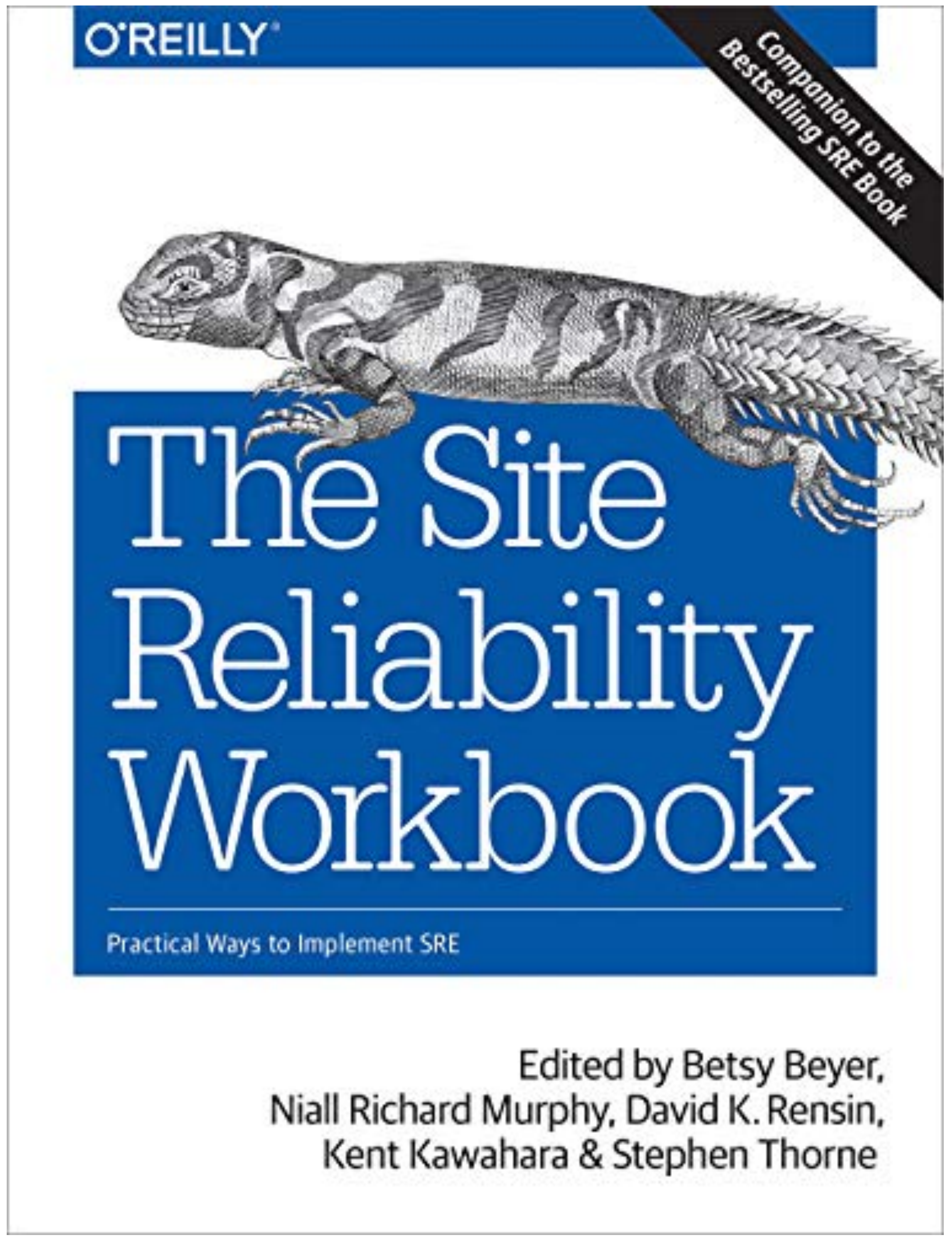
SRE at Google

Published two books

The original "blue bible"

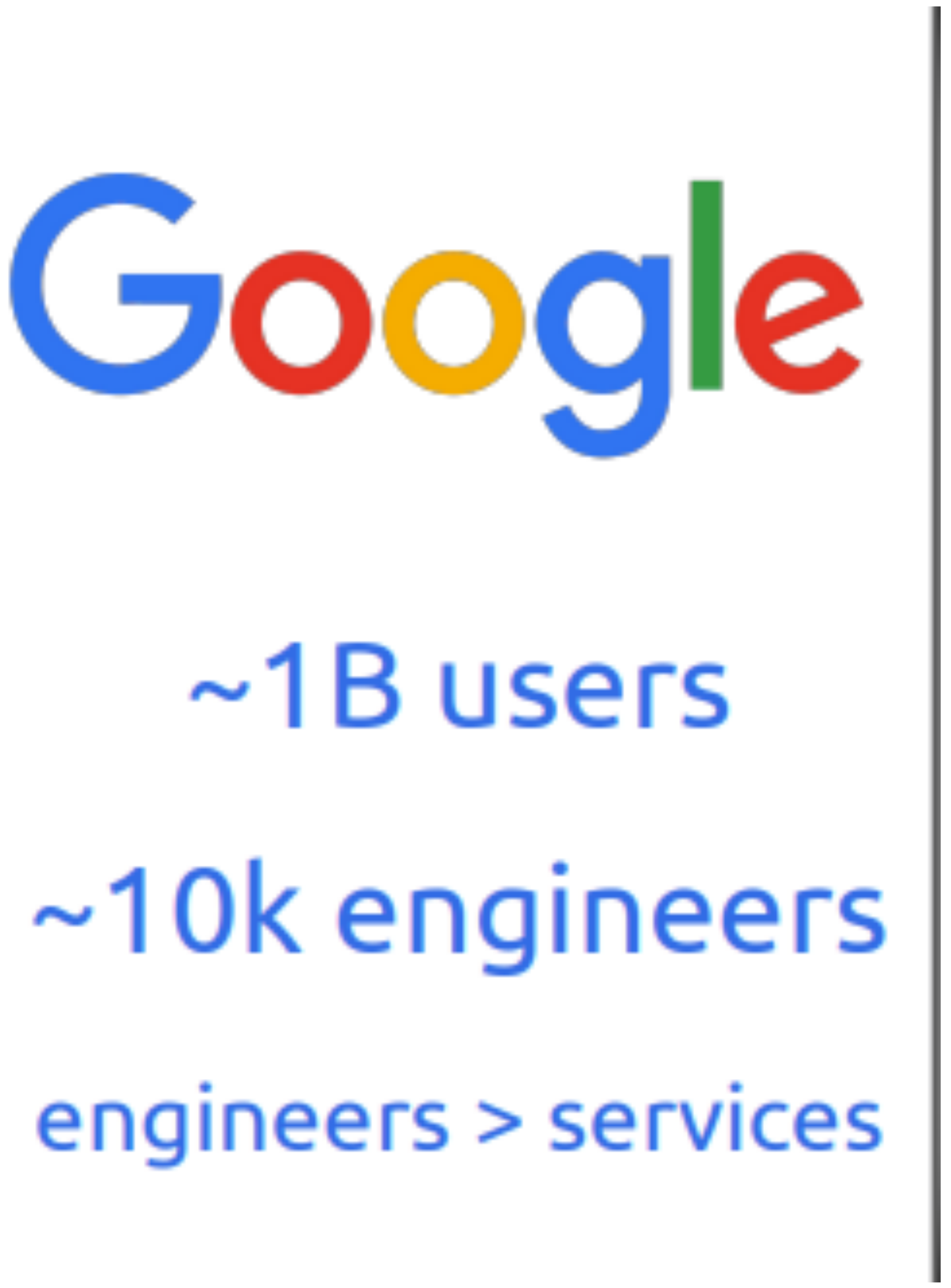
The workbook (experiences with CRE - Customer Reliability Engineering)


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Richard Murphy


O'REILLY
Companion to the Bestselling SRE Book
The Site Reliability Workbook
Practical Ways to Implement SRE
Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara & Stephen Thorne

Reminder:

You are not Google



Google

~1B users

~10k engineers

engineers > services

Source: Björn Rabenstein, SREcon

SRE at Google

- Google has an SRE and a DEV organisation.
- SREs are embedded in DEV teams
- SREs have SLIs/SLOs, can push back and make tomorrow better than today

Service Level .*

- Service Level Indicator (SLI): sensor, gauge
- Service Level Objective (SLO): expectation
- Service Level Agreement (SLA): contract

Service Level .* Consequences

- SLIs require monitoring
  - Client side instrumentation / EUM
  - Server side request logs/metrics
  - Front end infrastructure metrics
- SLOs require understanding the customer needs
  - Hard question
  - Incremental approach
  - Meaningful, e.g. what means availability in a Microservices architecture?

Error Budget

- Error Budget = 1 - SLO
- SLO = 99,9% availability
  - => Error Budget = 0.1% allowed downtime/failed requests
- or: SLO = 99,9% of requests are faster than 150 ms in the 95th percentile

Risk and Error Budgets

SRE implements DevOps

Source: SRE course, Coursera

SREs can say “no”

- Error Budget spent: no launches until issues are fixed
- SREs can return the pager to the DEV team
- SREs can leave a DEV team without consequences
- Ability to create back pressure makes a self-regulating loop
  - —> Removes major conflict between DEV and OPS

Make tomorrow better than today

- SREs are coders
- 50% cap on ops work
  - Ops work above those 50% will be assigned to DEV team
  - Self-regulating, DEV team sees system in action
- 50% dev work: write software to reduce “toil”

Ops Team

Alone on call
Fix all the mess
Stakeholder

SRE Team

On call with devs
Push back
Part of dev team

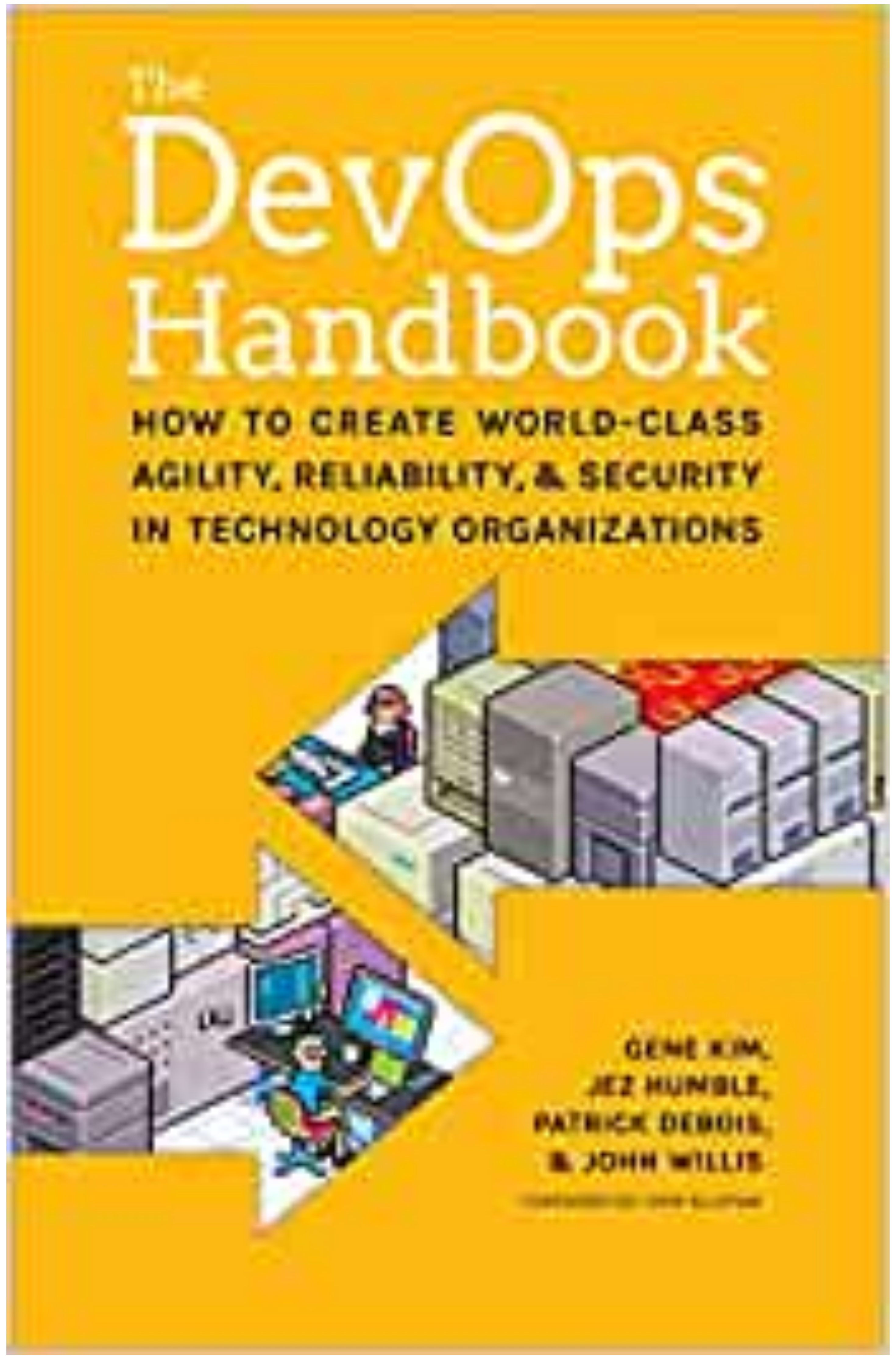
Is SRE an Ops Replacement?

- SRE balances feature velocity and stability
- Systems without feature velocity likely do not need SRE practices
  - On premise data center
  - Packaged software

SRE and DevOps

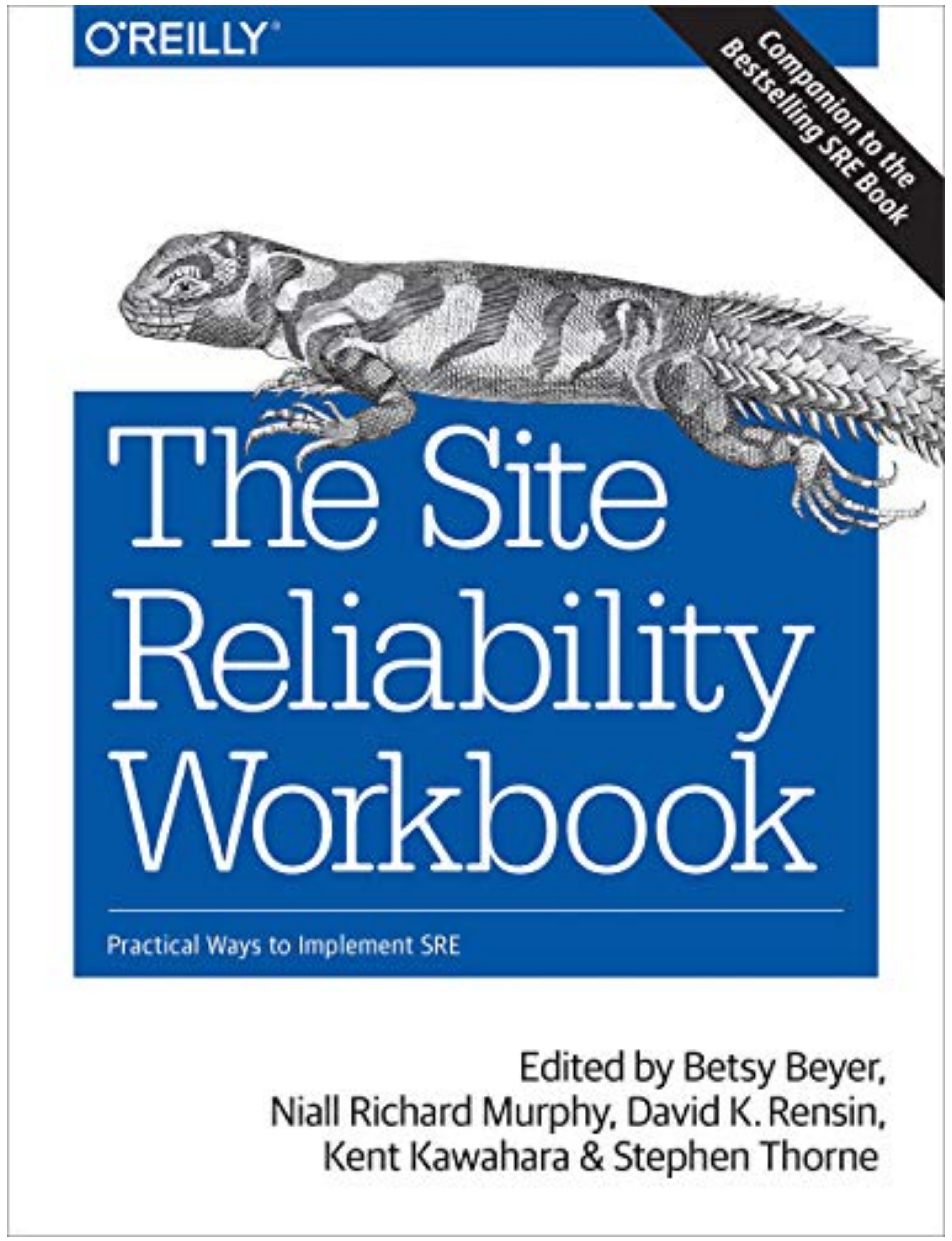
DevOps

- Break down silos between dev, ops, security and biz
- Accidents are normal (focus on MTTD/MTTR and change fail rate)
- Change is gradual (CI, CD)


The DevOps Handbook
HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS
GENE KIM, JEZ HUMBLE, PATRICK DEBOIS, & JOHN WILLIS

SRE

- Manage by SLOs
- Minimize toil
- Automate this year's job away
- Share ownership with developers


O'REILLY

Companion to the Bestselling SRE Book

The Site Reliability Workbook

Practical Ways to Implement SRE

Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara & Stephen Thorne

Commonalities

- SRE's effective shared ownership and DevOps' collaboration model
- Change is best pursued in small, continual steps
- Right tooling is really important, but tools don't tell you if you achieved something
- Measurement is key
- Shit happens in prod - practice blameless postmortems

DevOps

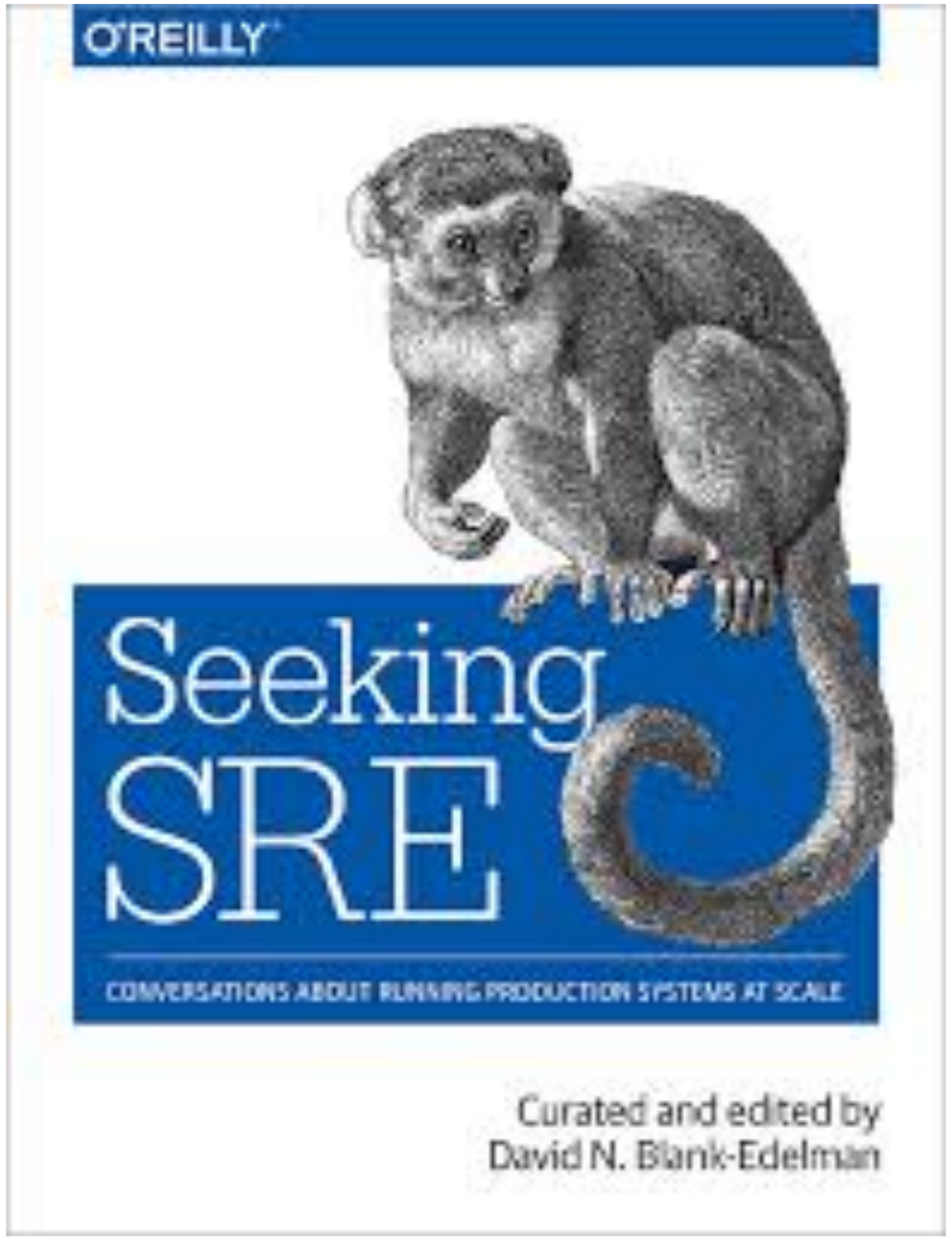
Wider Philosophy
Whole business
Silent on how to run ops

SRE

Narrow roles
Service oriented
Framework on how to run ops

SRE for non-Googlers

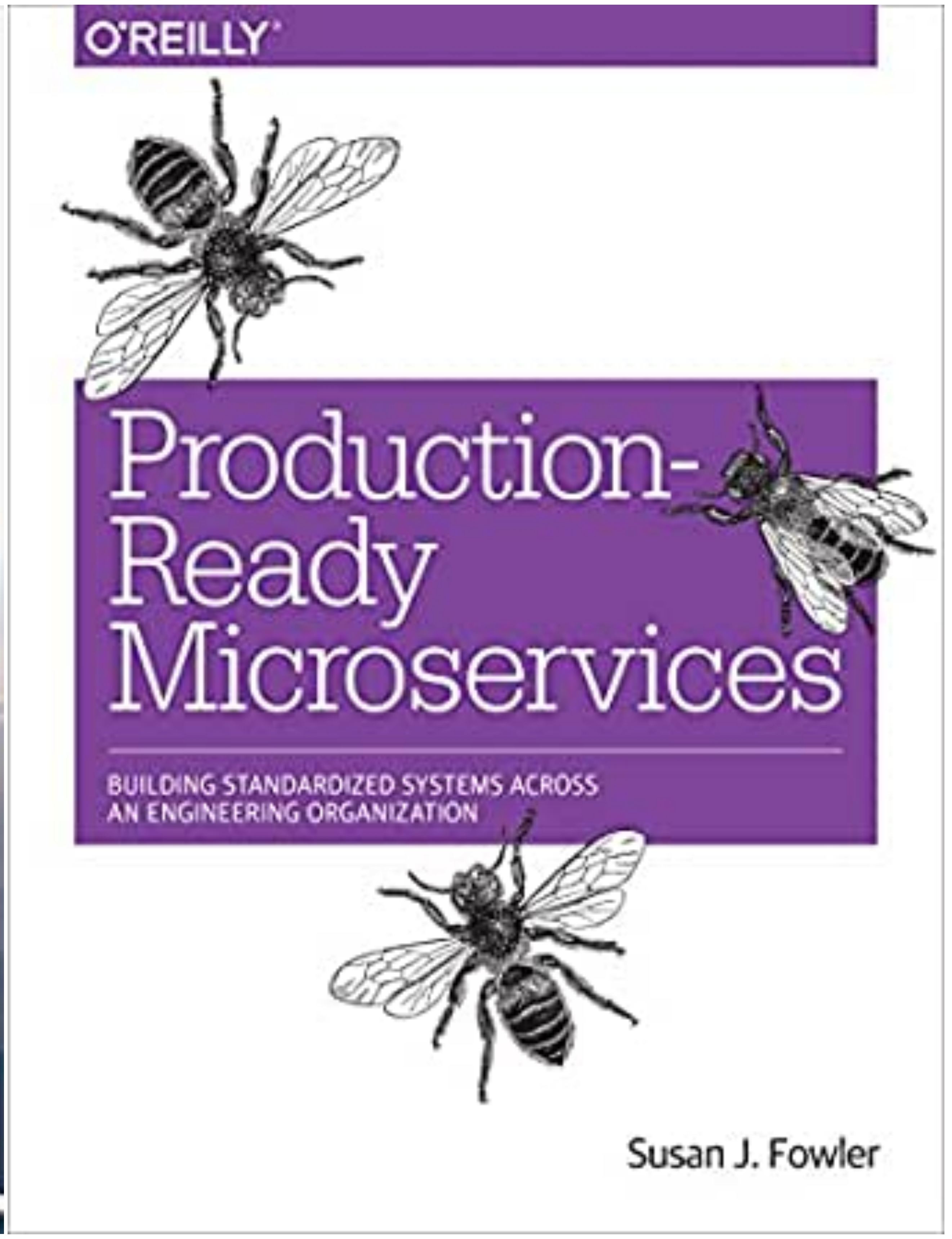
- "Seeking SRE" collects interesting insights how companies adopt SRE
- YBIYRI with SRE support looks promising
- "SRE in Spirit"


O'REILLY
Seeking SRE
CONVERSATIONS ABOUT RUNNING PRODUCTION SYSTEMS AT SCALE
Curated and edited by David N. Blank-Edelman

YBIYRI and SRE

- Small size: have ops/prod skills in the team
- Team with strong dev and ops skills supporting dev teams
- Trainings
- Reviews
- Checklists
- Support
- Templates
- Join production and fix the mess

The Pragmatic Programmers
Release It!
Second Edition
Design and Deploy Production-Ready Software
Michael T. Nygard
Edited by Katharine Dvorak


O'REILLY
Production-Ready Microservices
BUILDING STANDARDIZED SYSTEMS ACROSS AN ENGINEERING ORGANIZATION
Susan J. Fowler

Thanks