



# How Google SRE and developers work together

**Jennifer Petoff, Christof Leng**

Twitter: @jennski, @moroquen

Linkedin: [www.linkedin.com/in/jpetoff](https://www.linkedin.com/in/jpetoff), [www.linkedin.com/in/christofleng](https://www.linkedin.com/in/christofleng)



Site Reliability Engineering

Hello  
my name is

Christof Leng  
Google Germany

- Leads the SRE Engagements team
- Co-author of the Google SRE engagement model
- 7 years at Google
- Ph.D. in Computer Science

Image: Public domain, [https://commons.wikimedia.org/wiki/File:Hello\\_my\\_name\\_is\\_sticker.svg](https://commons.wikimedia.org/wiki/File:Hello_my_name_is_sticker.svg)

Hello  
my name is

Jennifer Petoff  
(aka Dr. J)  
Google Ireland

- Ph.D. in Chemistry
- 14 years at Google
- Lead the SRE EDU team
- Co-editor of the SRE Book
- Part-time Travel Blogger at Sidewalk Safari

Image: Public domain, [https://commons.wikimedia.org/wiki/File:Hello\\_my\\_name\\_is\\_sticker.svg](https://commons.wikimedia.org/wiki/File:Hello_my_name_is_sticker.svg)



2+ billion  
lines of code

Google's production environment might be one of the most complex integrated systems humanity ever created.

# How to run a planet-scale system?



“

SRE is what happens when  
you ask a software engineer  
to design and run operations.”

**Benjamin Treynor Sloss**, Vice President of 24x7 Engineering, Google



# We need specialists to design and run our systems



## Reliability

Meet the availability targets the users need.



## Velocity

Maximize the long-term feature velocity.



## Maintainability

Use software rather than human toil.



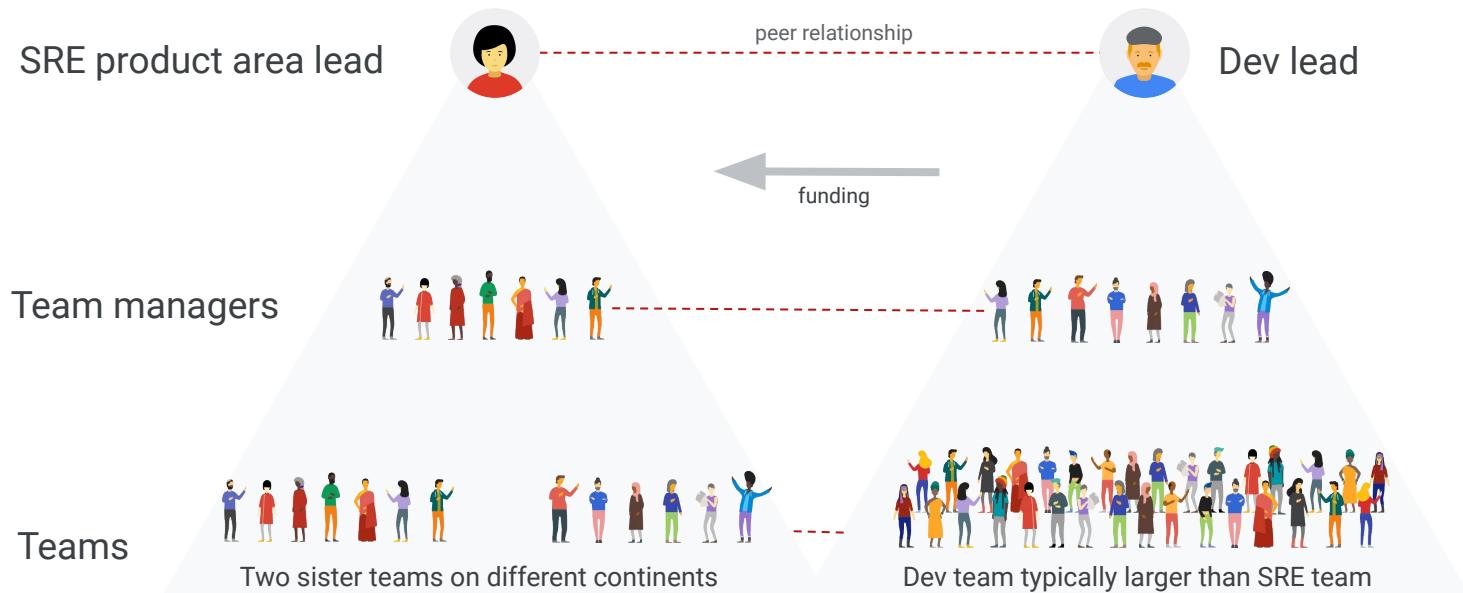
## Efficiency

Use engineering time and machine resources efficiently.

# SRE works on many different Google products



# SRE is its own organization that partners with Dev



# What is an SRE engagement?



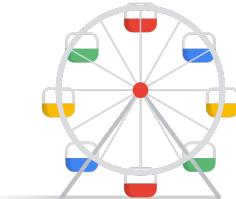
Peer relationship between  
SRE & Dev team



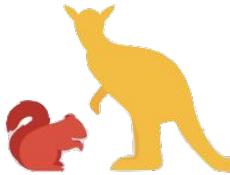
Scoped around specific  
service or product



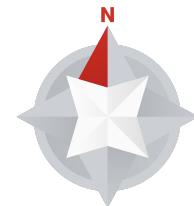
Requires contributions  
from both sides



Can span any part of the  
service lifecycle



Takes many different forms



Aligned with the SRE principles

# The SRE Engagement Model

# SRE support is not automatic



Funded by Dev



Strategic Partnership



Dev Ownership



Consensual Partnership



Funding conversations are typically led at the director/VP level.

# What should SRE work on?



Align with  
SRE's mission



Clear  
value proposition



SRE is not an  
“ops team”



Ops is not a  
zero-sum game



Oncall coverage does not justify the formation of an SRE team. Focus on sustained engineering value.

# How to make SRE work impactful



Shared Endeavor



Success  
must be tracked



SLOs and  
Error Budgets



Adjust investments



Roadmaps

SRE and Dev should maintain a shared roadmap for the engagement with clear success criteria.

# Focus on the important stuff



Clear scope



Advocate for the user



Promote standardization



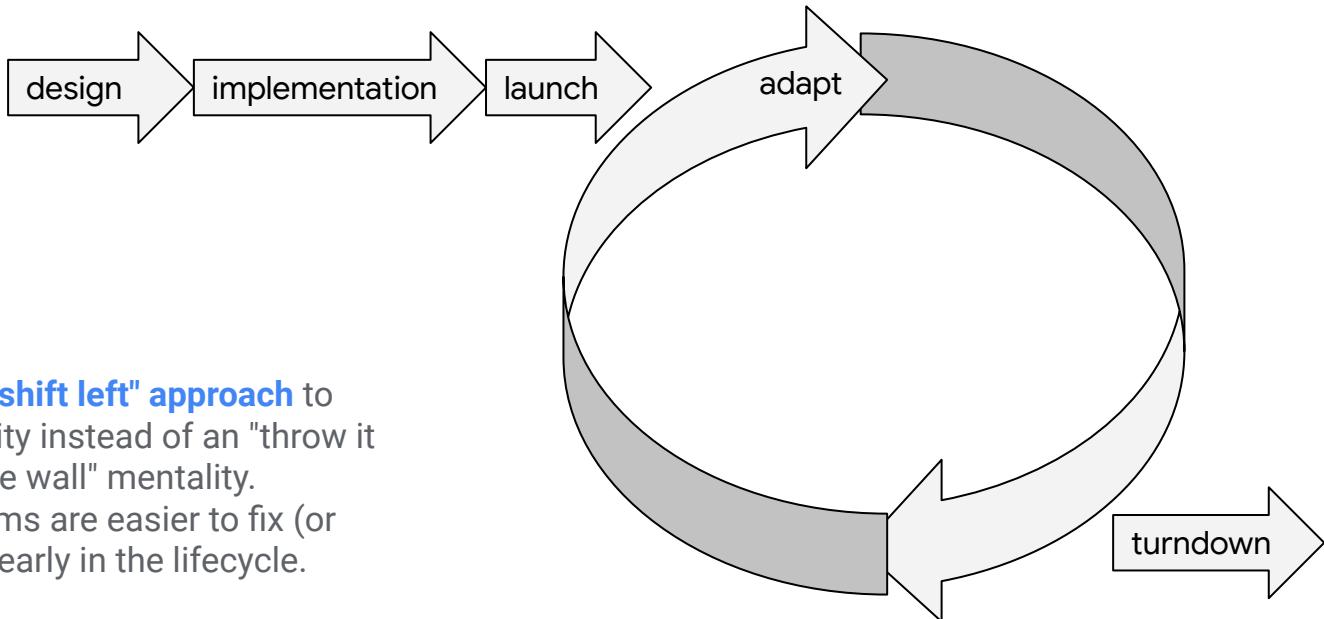
Teach to fish

# Engagement Types

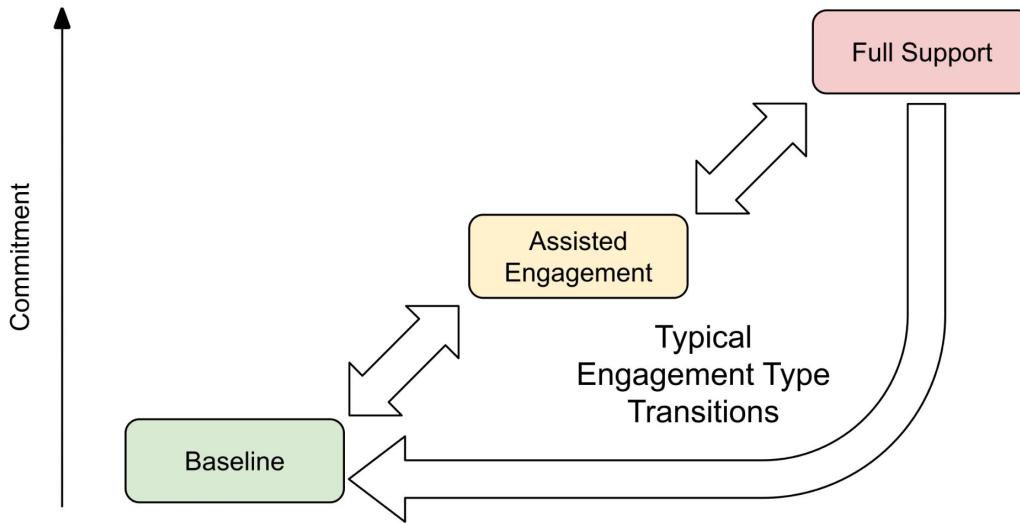
# Engage throughout the Service Lifecycle



Use a **"shift left"** approach to reliability instead of an "throw it over the wall" mentality. Problems are easier to fix (or avoid) early in the lifecycle.



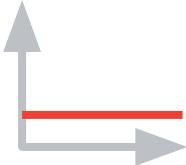
# Engagements can be classified into three broad types



Adjust to your needs

There is no one-size-fits all mix of engagement types or lifecycle phases. Use what works best for your situation.

# Baseline



Dev should have access to **base-level, ad-hoc support** when needed. This is the default engagement type in the absence of a specific service engagement type (in which headcount would be a consideration).



**SRE office hours** or fixed-time **consulting projects** to get production advice and technical design feedback.



Access to an **incident response team (IRT)** for coordinating large-scale outages. Cannot provide service-specific advice, but can help with communication and generic production expertise.

# Baseline Example: SRE Love



Time-boxed projects



Concrete proposal and  
deliverables defined by Dev



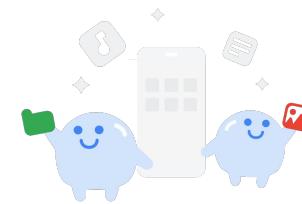
Project is run by Dev with  
guidance from SRE



Knowledge transfer



Helping productionization  
early in the lifecycle



Building a Dev/SRE  
relationship

# Assisted Engagement



SRE provides strategic, proactive, project-focused consultancy to the Dev team. There is a dedicated **SRE point of contact** for the service and a **shared roadmap** for the production aspects of the service.



**Joint projects** between SRE and Dev to improve the service health. Production is still owned by Dev, but SRE sometimes participates in a **shared oncall** to get a deeper understanding of the service.



SRE may take on roles such as **co-design, applying reliability expertise and experience**. Such engagement at an early stage (e.g. co-design) is one of the most highly leveraged (and thus valuable) activities SRE can undertake.

# Assisted Support Example: Embedded SRE



SRE(s) temporarily join Dev team for project work



Expensive: Only for highly critical projects



SRE principles during design & implementation

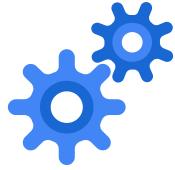


SRE must be adding more value than extra Devs would



Working towards a major launch or SRE onboarding

# Full Support



Full Support is the **highest commitment** support type, in terms of headcount funding, time investment from the Dev team, and SRE authority over the service.



It typically entails **SRE oncall responsibilities**, and requires a model of **shared ownership** and close coordination between both sides.



**Significant project work from both sides** is expected to fully comply with SRE best practices. The focus of this work is to reduce the long-term effort and risk of operating and maintaining the service.

# Full Support Example: Automating yourself out of your job



Redesign a service's infrastructure or architecture



Reduce the need for continuous SRE involvement



Knowledge of the service and its problems needed



More time for other services or engineering projects



Reduces cost of complex, high-touch systems



Best suited for mature and business critical systems

# What if everything goes wrong?



Operations overload

Disagreement on direction

Lack of engagement

Google



Find the root cause

Declare "code yellow" for priority work

Escalate up the management chains

Don't be a hero



Remember the engagement principles

Repair the relationship

You can only be successful together

Site Reliability Engineering

# Summary

# Google SRE in a nutshell



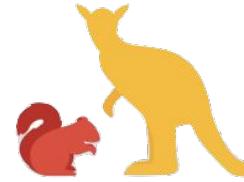
SRE is a specialist organization

- with a principled approach
- to balance reliability and velocity
- with maintainability and efficiency in mind



SRE partners with Dev

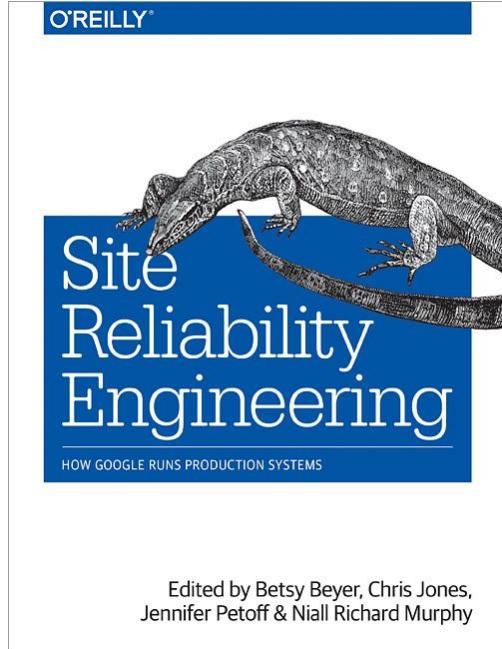
- to solve hard engineering problems
- for shared success
- to increase production knowledge throughout the service lifecycle



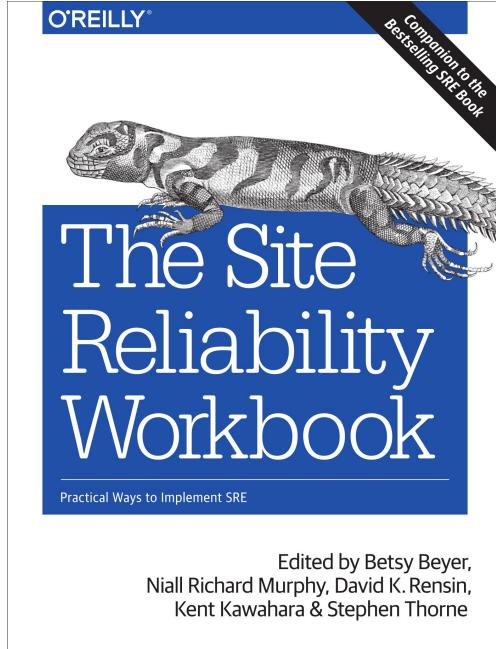
SRE engagements

- take many forms
- need to fit the needs of the service
- evolve constantly

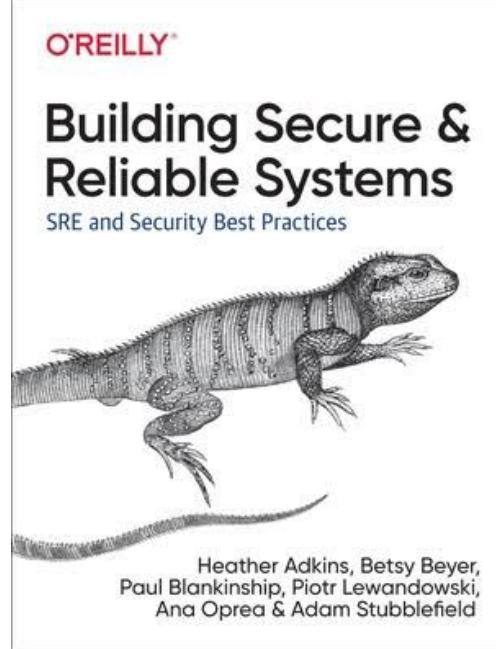
# O'Reilly Books



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



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



Heather Adkins, Betsy Beyer,  
Paul Blankinship, Piotr Lewandowski,  
Ana Oprea & Adam Stubblefield

Read online at <https://sre.google>

# We'd like to engage with you (pun intended)



How does SRE work at your organization?

- If you have separate Dev and SRE orgs, what best practices have you found in working together?
- If not, how do you ensure that balance between reliability and velocity?



What other SRE topics would you like to hear about?

- Check out [sre.google](https://sre.google) and [cloud.google.com/sre](https://cloud.google.com/sre) for our current publications.



Connect with us!

- Twitter:  
[@jennski](https://twitter.com/jennski)  
[@moroquen](https://twitter.com/moroquen)
- Linkedin ([linkedin.com/in/...](https://linkedin.com/in/jpetoff))  
[jpetoff](https://linkedin.com/in/jpetoff)  
[christofleng](https://linkedin.com/in/christofleng)



Thank you!