Google

Overcoming challenges for a successful SRE organizational setup

Enterprise Technology Leadership Summit Europe 2024

Christof Leng

Site Reliability Engineering

DevOps

- Running software in production is an essential part of the software development lifecycle
- Everyone should have production knowledge, break down silos
- That is definitely a great idea (every org should do it)
- All you need when your production complexity is limited (e.g. startup, small services)

Photo by Amal Abraham

But what happens as complexity increases?

Photo by Monstera Production

- Software engineers must have knowledge of many domains
  - Deep understanding of the business domain
  - Many horizontal topics (UX, testing, security, etc.)
  - Production knowledge is adding to the cognitive load
- You can't master everything
- You still need specialists when production is a complex and critical topic
- But how do you organize these specialists?

Embedded production experts?

- "Obvious solution": Have 1-2 full-time or part-time production experts per team or group of teams

Problems:

- The team may offload the grungy ops work to these poor folks
- The experts may be pulled into other work, leaving no time for proactive production work
- It's hard to establish a community for the dispersed experts. That makes it hard to grow the expertise and develop senior experts in the space.
- The role is a career dead end. Ambitious engineers will not bite.

Photo by Nida Kurt

Photo by NastyaSensei

A community of experts

- Teams of production experts (i.e. Site Reliability Engineers) can help to overcome these problems

Depending on the organization, it can be

- teams embedded individually in the Dev org
- an org of multiple teams for a bigger business unit
- a company-wide center of excellence

Read more at https://sre.google/workbook/team-lifecycles/

Organisational Challenges & Remedies

Antipattern #1: No business alignment

SREs in a separate team may not (fully) understand what is most important for the business and prioritize projects that are irrelevant or even getting in the way.

Solutions:

- Expose the SREs to the product and its users
- Have joint planning and projects between SRE and Dev
- Spend face-to-face time with Dev and product management
- Focus SRE engagement around end-to-end user journeys instead of the internal system structure
- Establish SLOs and OKRs that align with user happiness and business success

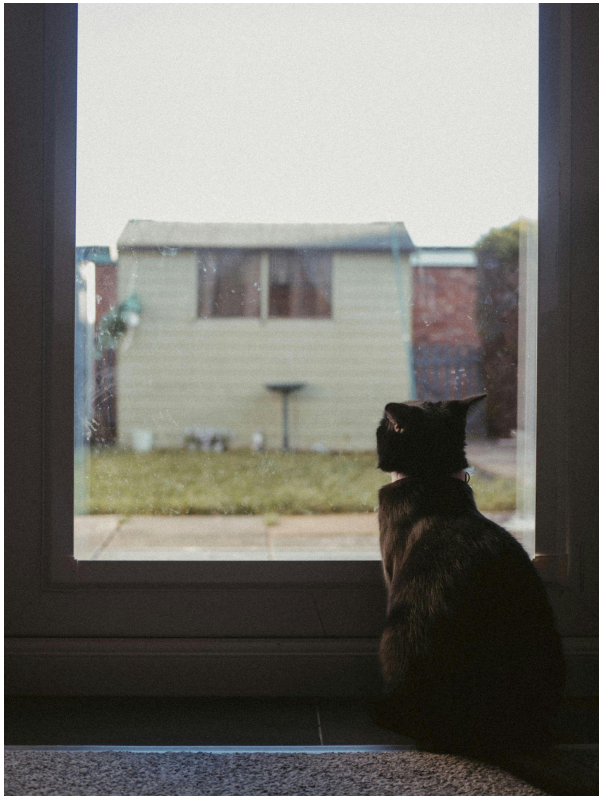

Photo by Lisa Fotios

Antipattern #2: Organizational Overhead

SRE is spread thin across many systems and has to interact with many stakeholders.

Solutions:

- Keep the space an SRE team operates in coherent.
- Limit the number of concurrent engagements.
- Only engage where the impact substantially outweighs the cost.
- Keep in touch with Dev, but make the meetings worth the time.
- In a global organization: Keep an eye on out-of-office-hours meeting load.

Photo by Nuhoglu Carpaner

Antipattern #3: SRE as an Ops Team

When SRE first and foremost means "being oncall for a system", there is a high risk of not having enough time and focus to make lasting engineering improvements.

Solutions:

- Give SRE access to the code and let them contribute.
- Only be oncall where it supports SRE's engineering roadmap.
- Return the pager to Dev when these goals have been achieved.
- Always share the oncall work with Dev.
- Define your engagements around engineering goals.
  Only discuss oncall once these goals are clear.

Photo by Arina Krasnikova

Antipattern #4: Career Dead End

SRE being perceived as a career-limiting or boring, limiting talent acquisition and retention.

Solutions:

- Involve SRE in strategic projects (e.g. new launches).
- Provide interesting work (design, coding, etc.).
- Provide sufficient career paths within SRE.
- Ensure SRE representation in promo decisions.
- Enable employee mobility between SRE and Dev.
- Invest in the skill growth of your SREs.

Photo by Lad Fury

Antipattern #5: Human Abstraction Layer for Production

The SRE tries to stay relevant by shielding Dev from production and building increasingly complex infrastructure that only they can maintain.

Solutions:

- Keep track of your production complexity.
- Reward the team for making production simple.
- Don't punish them for being successful.
- Define their role as a "sports coach", not as a "laundry service".
- Provide work more attractive than being oncall or building yet another piece of infrastructure.

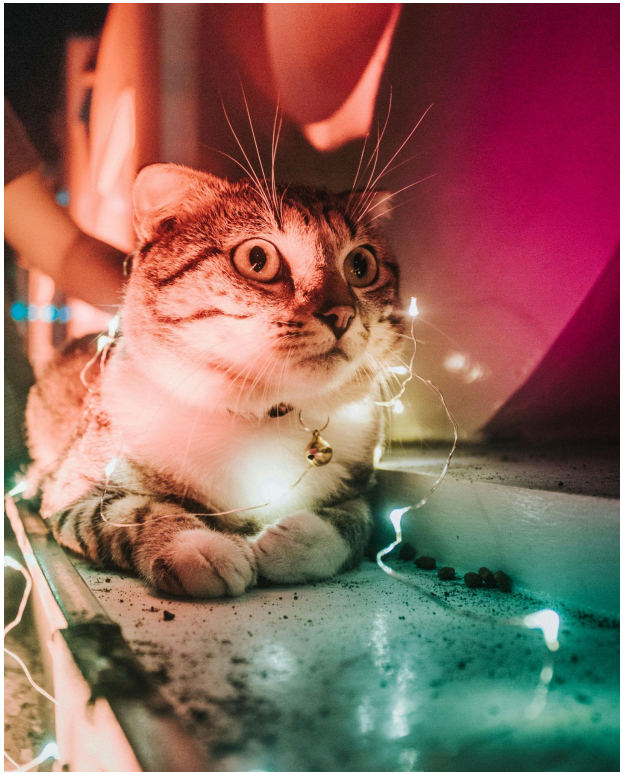

Photo by Carms Onoya

Wrap up

Summary

- Breaking down silos is critical.
- You may still need full-time experts.
- Provide an environment that they can thrive in.
- Help them to align their work with your business goals.
- Emphasize SRE/Dev collaboration throughout the product lifecycle.

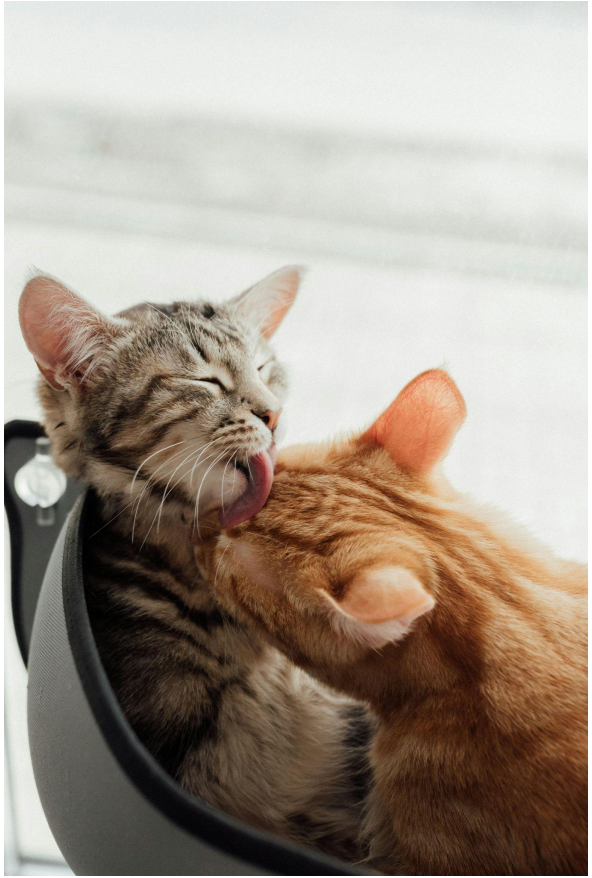

Photo by Arina Krasnikova