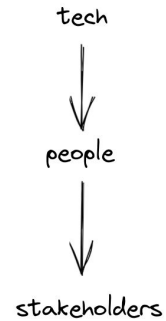


Oncall: An Equal-Opportunity Waste of Time

Dave O'Connor, Twilio (@gerrowadat)

buh

- Google SRE, 2004-2021
 - (2004) Systems Administrator II
 - (2021) Director, Storage & Databases SRE
 - Oncall for GMail, Google Analytics, BigTable, Storage
- Elastic SRE, 2021-2022
 - Sr. Director, Elastic Cloud SRE
- Twilio SRE, 2022-
 - VP Engineering, SRE Lead



I used to do tech now now I do stakeholder management. My life is a rollercoaster.

y u troll us

- is fun
- works

Much of the content here may not apply to you. I consider that a vindication of how varied our profession is, because it certainly applies to some of us.

Regardless, the intention here is to evoke thought and discussion; maybe by trolling y'all a little bit.

The Reward for Good Work is More Work

The Job of any engineer is to find a better class of first-world problem. This is especially true of SRE. We even say cute phrases like 'automate yourself out of a job every 18 months'.

However, just as someone grows their career and matures in their approach, SRE has also done this. One of the ways this happens is by thinking of things less as a band-aid on top of toil, and more as an engineering approach; part of that is putting aside less worthwhile things.

We're not going to eliminate oncall entirely. Shit happens. At some point we have to acknowledge that who covers the oncall is not that important.

If covering oncall continues to be seen as part of what SRE is special, we will continue to be seen as magical pixies who alone knows the spells to do things with production. That knowledge will by definition remain obscure. It will be our reward for being good at this.

Think about what you want to be known for being good at. Eliminate other work.

Oncall is a Waste of Time

(tax returns as a good use of time)

There's a huge difference between 'a good use of time' and 'a good idea'. It's a good idea to file taxes. It's not necessarily a good use of time.

Do you want to be super good at doing individuals' taxes? That's a weird thing to want, especially if you're an economist as your main job.

Doing a tax return is an intentional result of policy. Your tax office could easily do them for you in the vast majority of cases.

The major difference is the mentality - for example, there's a whole industry and lobby in the US around disincentivising making tax filing easier.

I'm still only kinda trolling when I say oncalt is a waste of time, because it mostly is - it's not going anywhere, but that doesn't mean we have to like it.

The Fetishisation of Incident Response

Oncall being complicated is a self-fulfilling prophecy (3 months to oncall; 'black belts', etc.). If we choose to gatekeep oncall in this way, it's because it's too complicated. This is intentional; it is accepted. People are often secretly proud of it.

(Somewhat google-specific) DiRT/Disaster recovery, an exercise in pageantry and cosplay.

Reinforces the 'only SREs can be oncall' trope. That's the case because you have insisted as much and tolerated complexity because you understand it. It's 'toxic exceptionalism'.

People do not need to 'feel the pain'. Pain is not weakness leaving the body. It is pain entering the body

To tie these things together – the attitude that only SREs can be oncall ties in with the notion that Oncall is a waste of time - Why is it them considered valuable?

Why We Invest in SRE

Going to get menagery for a bit here. Stakeholder management.

At some point behind the scenes there's someone deciding that adding an SRE is a better idea than adding a product SWE. What do they think about?

I speak with a lot of these people. There is a strange easy/hard-path when we look at SRE. People often ask for more SREs because they want them to cover more oncall, or more technical debt. Even if we pay SREs the same, there's still value in an organisational separation where one group does product and the other operations. It's a very tempting easy path to take.

The harder path is to really look at whether the engineering output of each engineer contributes to delivery in a way that's optimal, or even multiplicative.

It's easy to say 'Follow the sun' and then think about what these people's actual jobs are later.

Losing the Training Wheels

How do we move beyond oncall/ops as a crutch for proving our value to stakeholders?

I'm not saying we should be bad at oncall or incident response – or that oncall should be dumbed down to the point of pointlessness. Each overall org has to continue to be capable of taking care of business when the unforeseen happens.

The key thing to look at is the 'training wheels' of there being an easy path to justifying your existence. How do we make sure we're really looking at how rare SREs are, and if they are of sufficient rarity? Getting to a place where we've really thought about the quality

Of each and every SREs engineering remit is difficult and can even feel self-sabotaging.

A Post-Magical Era

Exercises

(1)

- With a group of leads, eliminate your 'slack' when it comes to oncall
 - Per-site minimums (by policy, or the Google-book 6 or 7)
 - Ops work target percentages (50% at G, likely less elsewhere).
- Right-size your project work to essentials

The catch – it's always easy to justify the existence of project work when the staff are already funded. Try to justify existing staffing levels assuming it's not staffed yet and you need new heads.

(2)

Imagine a deranged exec has demanded that all oncall must be shared co-equally by engineer.

What parts do we still expect engineers to be good at? How do we compartmentalise expertise?

What aren't we doing that already?

The SRE Value Proposition

Slide included to raise the blood pressure of Google SRE leadership.

For everyone else it's a call to raise all boats. Muscle memory requires exercise, and having your ass saved by SRE every so often often isn't enough to shake the prevailing view of SRE as fancy-ops. It needs to be clear in people's minds what we're there for.

On stakeholders - moving the needle will be hard. Part of removing the base assumptions that SRE will cover your ops burden is to make it more normal for that to not be a thing. That exec wasn't deranged – they understood that if you launch a service people use, there's a collective responsibility to keep it going that you don't get to outsource. Normalise valuing our time more, and our remit will improve.