"We want to adopt SRE best practices"

A Lightning Talk

Steve McGhee
Reliability Advocate, SRE
Google

These are my opinions

not official guidance from Google :)

It sounds like a good question

Everybody asks this

Should be easy, right?

"Just send us a quick link!"

We
want to
adopt
(SRE / reliability / resilience)
best practices

We

Who exactly?
The business?
Leadership?
One engineer?
One team?

want to

"want" or "must"?

is there an existential need?

was there a bad outage?

is this someone's pet project?

what level of investment?

does anyone disagree / not want to? who/why?

any alternatives considered?

adopt

how do you expect adoption to happen?

be very specific:
"adoption of [SLOs] by 75% of teams on top 3 products"

how will you know it's working?
how will you know when you're done?
(hint: you're never done)

(SRE / reliability / resilience)

what are you actually trying to accomplish?

not all services require high reliability

computer systems can't be resilient without humans, as they can't adapt to new risks

Why SRE, specifically? ("we are not Google")

best practices

"a person asking for a best practice
doesn't know what they're asking for.
they're asking for someone else to define it."

there are no global best practices, for all systems, for all teams

→ there are principles ←

(principles require interpretation, sorry)

TL;DR:
this is the **wrong path.**

instead:

adaptability > **prescription**

"help me understand"

"tell me how to do it"

"*best practices*, once established **can't be argued**,
... if you **don't** follow them and a problem happens, you're in trouble.
... if you **do** follow them, you're still in trouble.
... transferring the **burden of deciding** to the practitioner."

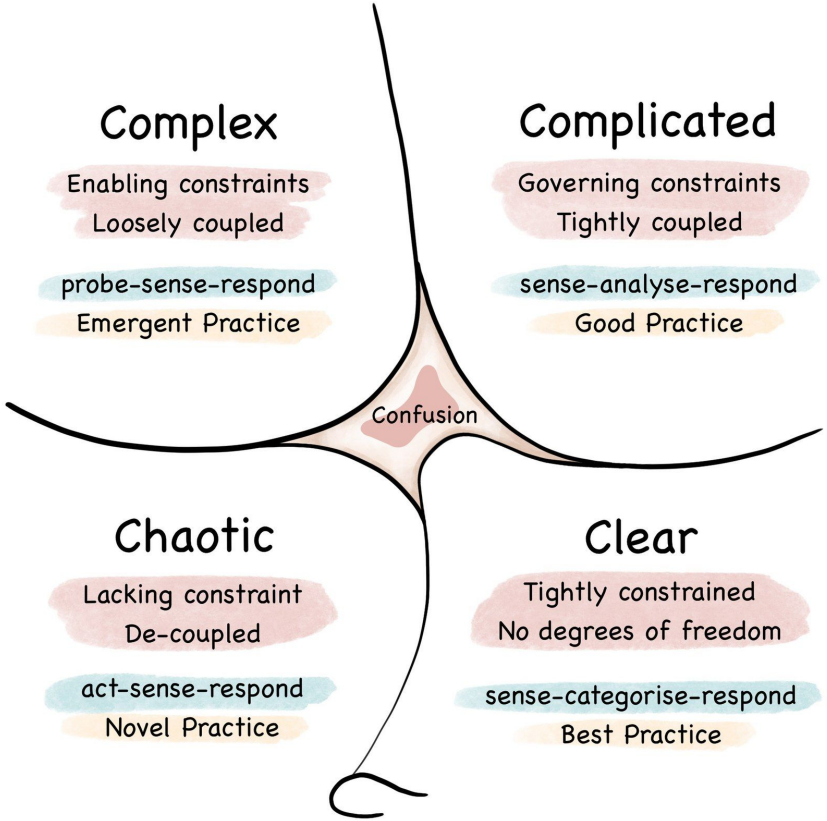
~ Dr. Richard Cook
https://www.adaptivecapacitylabs.com/blog/2022/09/12/richard-cook-a-life-in-many-acts/

It is important to point out that this is all in the context of Cloud Computing.

Best Practices make perfect sense in other contexts.

In Cloud Computing, especially in large Enterprises, we are working with Complex systems.

One model for interpreting this is the **Cynefin** framework →



https://en.wikipedia.org/wiki/Cynefin_framework#/media/File:Cynefin_framework_2022.jpg

what you really need is the **ability to learn about** and **adapt to** a complex system.

best practice **isn't** around deployment patterns or tools,
it's about **honest introspection**,
failure analysis,
and **sustainable prioritization of work**.

You will know it's **working** when…

"… yesterday's best practices become **questioned** and eventually **revised** or even **discredited**."

"In complex situations, heuristics and pattern recognition are essential and **cannot be replaced by sets of rules."**

Can We Trust Best Practices? Six Cognitive Challenges of Evidence-Based Approaches

Klein, Devorah & Woods, David & Klein, Gary & Perry, Shawna. (2016). Can We Trust Best Practices? Six Cognitive Challenges of Evidence-Based Approaches. Journal of Cognitive Engineering and Decision Making. 10. 10.1177/1555343416637520.

You have to **learn** which parts are complex, then simplify them,

learn how they fail,
then mitigate those failures.

then repeat, forever.

https://how.complexsystems.fail/
https://www.learningfromincidents.io/

You **won't know** until you do it.

You won't know until **you** do it.

You won't know until you **do it**.

You won't know until you do it.

FIN

Sources for the concepts presented here:

- Rethinking Best Practices – Will Gallego
- Brendan Crake - Banning “Best Practice”
- The Cynefin Framework
- What Safety-II Isn’t – Humanistic Systems
- The Varieties of Human Work – Humanistic Systems
- Efficiency-Thoroughness Trade-Off - ETTO principle - erikhollnagel.com
- Can We Trust Best Practices? Six Cognitive Challenges of Evidence-Based Approaches
- Best Practices Are Stupid 40 Ways to Out-Innovate the Competition

“In a complex system, there is no such thing as ‘best practice’, except perhaps **practice that is best at a certain time in a certain context** – the best that could reasonably be done given the demand, resources and constraints. The best practice we can hope for is **contextual practice** – **practice that fits the context**.”

Dr. Steven Shorrock
What Safety-II Isn’t – Humanistic Systems

When things are **Simple**: Best practice ← are you **really** here?

You should have playbooks,
You should take backups,
You should monitor stuff, test stuff, version your configs, …

"I am once again asking you to put things in multiple cloud zones"

More examples of those: (Architecture Guides, github, etc)
you can probably copy them with minimal modifications,
unless things are getting **Complicated…**

Complicated: "Good" practice

Auto failover is good,

- **but** you need to make your system stateless,
- so **maybe don't always** auto failover,
- unless you really **want to?**,
- **but** stateless is hard,
- either way **check with an expert** on this topic :(
- Unless the expert hasn't seen this **exact setup** before,
- or if you're doing this for the first time

So, maybe things are getting **Complex...**

Complex: Emerging practice

The first time you launch your services you have **all the simple and complicated things covered**, but then in the **real world** launch you find out a load of **emergent things that were impossible to predict**.

So you **run a retrospective**, do more tests, add the learning, and hopefully make the complex things just **complicated** or maybe even **simple**.