Building Successful SRE in Large Enterprises

Dave Rensin
Director, Customer Reliability Engineering & Global Network Capacity Planning
@drensin // rensin@google.com

Liz Fong-Jones
Staff Developer Advocate for SRE
@lizthegrey
Reliability is the most important feature.
Our users measure our reliability.
How do we improve reliability?

DevOps? or SRE?
The Principles of DevOps

- Reduce organizational silos
- Accept failure as normal
- Implement gradual changes
- Leverage tooling and automation
- Measure everything
The Key Principle of SRE

“100% is the wrong reliability target for basically everything.”

Benjamin Treynor Sloss
Vice President of 24x7 Engineering, Google
<table>
<thead>
<tr>
<th>Availability level</th>
<th>Allowed unavailability window</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>per year</td>
</tr>
<tr>
<td>90%</td>
<td>36.5 days</td>
</tr>
<tr>
<td>95%</td>
<td>18.25 days</td>
</tr>
<tr>
<td>99%</td>
<td>3.65 days</td>
</tr>
<tr>
<td>99.5%</td>
<td>1.83 days</td>
</tr>
<tr>
<td>99.9%</td>
<td>8.76 hours</td>
</tr>
<tr>
<td>99.95%</td>
<td>4.38 hours</td>
</tr>
<tr>
<td>99.99%</td>
<td>52.6 minutes</td>
</tr>
<tr>
<td>99.999%</td>
<td>5.26 minutes</td>
</tr>
</tbody>
</table>
Error budgets

- Product management & SRE establish an **availability target**.
- 100% - availability target is a “budget of unreliability” (or the **error budget**).
- Monitoring measures **actual uptime**.
- Control loop for utilizing budget!
**Glossary of terms**

**SLI**
- **service level indicator**: a well-defined measure of 'successful enough'
  - used to specify SLO/SLA
  - $\text{Func(metric)} < \text{threshold}$

**SLO**
- **service level objective**: a top-line target for fraction of successful interactions
  - specifies goals $(\text{SLI} + \text{goal})$

**SLA**
- **service level agreement**: consequences
  - $\text{SLA} = (\text{SLO} + \text{margin}) + \text{consequences} = \text{SLI} + \text{goal} + \text{consequences}$
The Practices of SRE

- **Metrics & Monitoring**
  - SLOs
  - Dashboards
  - Analytics

- **Capacity Planning**
  - Forecasting
  - Demand-driven
  - Performance

- **Change Management**
  - Release process
  - Consulting design
  - Automation

- **Emergency Response**
  - Oncall
  - Analysis
  - Postmortems

- **Culture**
  - Toil management
  - Engineering alignment
  - Blamelessness
Why not both? SRE implements DevOps

- **Reduce organizational silos**
  - Share ownership

- **Accept failure as normal**
  - Error budgets & blameless postmortems

- **Implement gradual changes**
  - Reduce cost of failure

- **Leverage tooling and automation**
  - Automate common cases

- **Measure everything**
  - Measure toil and reliability

@lizthegrey & @drensin at #VelocityConf
About us

Liz Fong-Jones
Staff Developer Advocate for Site Reliability Engineering, Google

Dave Rensin
Director, Customer Reliability Engineering; Director, Global Network Capacity Planning, Google
Why Enterprises ❤️ SRE
Enterprises understand TCO and ROI

Run time >> development time.

Error budgets and SLOs prevent “intuition fatigue.”

Tools to both go faster and be more reliable un-paint executives from their corners.
Enterprises appreciate cost savings.

Not just in dollars -- agility and opportunity costs.

Incentives to **reduce complexity**.

Space for innovation matters.
SRE manages risk.

SRE philosophy quantifies and mitigates risks

Regulated industries have audit and inspection requirements:
- Financial Services
- Healthcare
- etc...

SRE unifies regulatory policy and operational principles.
“You should describe how and when risk analysis was or will be performed. Your design validation procedure(s) should describe how you will document, use, and update your risk management program. For additional guidance on risk analysis and risk management activities, see the QS regulation preamble comment #83. [61 FR 52620-52621; see footnote 2.]”

-- Quality System Information for Certain Premarket Application Reviews; Guidance for Industry and FDA Staff

@lizthegrey & @drensin at #VelocityConf
The FDA requires risk analysis!

<table>
<thead>
<tr>
<th>Risk</th>
<th>Bad minutes/year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overload results in slow or dropped requests during the peak hour each day.</td>
<td>3559</td>
</tr>
<tr>
<td>A bad release takes the entire service down. Rollback is not tested.</td>
<td>507</td>
</tr>
<tr>
<td>Users report an outage because of network problems.</td>
<td></td>
</tr>
<tr>
<td>There is a physical failure, and restoration from a backup is not tested.</td>
<td></td>
</tr>
<tr>
<td>The wrong server is turned on.</td>
<td></td>
</tr>
<tr>
<td>Overload results in a cascading failure, and system operators are not trained.</td>
<td></td>
</tr>
<tr>
<td>Operator accidentally deletes the wrong file.</td>
<td></td>
</tr>
<tr>
<td>Unnoticed growth in usage of the application.</td>
<td></td>
</tr>
<tr>
<td>A configuration mishap results, for example, in unhandled requests.</td>
<td></td>
</tr>
<tr>
<td>A new release breaks a small number of requests.</td>
<td></td>
</tr>
<tr>
<td>Operator is slow to debug issues.</td>
<td></td>
</tr>
<tr>
<td>A daylight savings bug drops requests.</td>
<td>71</td>
</tr>
<tr>
<td>Restarts for weekly upgrades drop in-progress requests (i.e., no lame ducking).</td>
<td>52</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SLO</th>
<th>99.9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Budget</td>
<td>525.6 min/yr</td>
</tr>
</tbody>
</table>

You can play with this tool yourself at: https://goo.gl/bnsPj7

The FDA requires risk analysis!
SRE can be an easier lift

SRE is a **concrete** set of practices.

SRE provides a **consistent** and **optimized** way of implementing DevOps principles.

Executives can quantify and measure benefits.
How to start with SRE
(0) Willingness is the thing

It doesn’t matter from where you start, as long as you’re willing to do the work you can do SRE.

The ops and dev talent in an Enterprise are up to the task -- just align the incentives

A company doesn't have to look anything like Google, Netflix, LinkedIn, etc to do it well.
(0) In Practice -- Anonymized

A customer tried adopting SRE without a clear executive sponsor; the sponsor churned 3 times and the project stalled.

This would have been more successful with a written plan for successors to **continually revise and execute** and let it grow organically.

**Note:** “Executive sponsor” != “Executive mandate”
(1) Do one application first

application
/ˌapləˈkæSH(ə)n/

Noun

noun: application; plural noun: applications; noun: application program; plural noun: application programs

1. A discrete failure domain
(1) In Practice -- Anonymized

An enthusiastic enterprise customer tried to transform whole org in place, and it was disastrous.

The best way to do this is one discrete failure domain at a time and let it spread organically. You can’t change an entire culture in one fell swoop.
(2) Start with the Error Budget

If you can convince the exec, dev, and ops teams to create and stick to Error Budgets, then the rest (pretty much) takes care of itself.
“Start the conversation from the point of view of your customers: what promises are you trying to uphold?”

“We kept our first pass simple by focusing on uptime. Using this simple first approach, we could clearly articulate what we were measuring, and how.”

“‘Perfect is the enemy of good.’ Even when SLOs aren't perfect, they're good enough to guide improvements over time.”

“We selected an initial SLO that covered most, but not all, user interactions, which was a good proxy for quality of service.”

-- Ben McCormack (VP Operations / Chief of Staff -- Evernote)
“[Before our] culture of SLOs, monitoring tools and dashboards were plentiful, but were scattered everywhere and didn’t track data over time.”
“We began troubleshooting at the user-facing service and worked backwards until we found the problem, wasting countless hours.”

“If a team needed to build a service, they wouldn’t know if the service they had a hard dependency on could support them. These disconnects caused confusion and mistrust.”

“Once SLOs were firmly cemented and effective automation and reporting were in place, new SLOs proliferated quickly. After tracking SLOs for about 50 services at the beginning of the year, by the end of the year we were tracking SLOs for 800 services, with about 50 new services being registered per month.”

-- William Bonnell (Sr. Director, SRE – The Home Depot)
(3) Alerting/Monitoring & Ops Load

TL;DR:

More logging and measurement is (probably) better;

More alerting is (probably) not!

Symptoms of pain, not infinite potential causes.

Focus on Observability.
(4) Blameless culture

We will always be reacting to the same kinds of failures over and over unless we invest in discovering what happened. We really ought to get something out of every error, rather than wasting the opportunity.

Can't get to culture of being able to take risks if we're blameful.

Blame guarantees deceit!

(see why at: https://goo.gl/RBdYwc or)
Don't try it all at once

One step at a time...

1. Define SLIs, SLOs, and Error Budgets
2. Audit and adjust monitoring and alerting
3. Model and reward blameless postmortems
Don't try it all at once

One step at a time…

1. Define SLIs, SLOs, and Error Budgets
2. Audit and adjust monitoring and alerting
3. Model and reward blameless postmortems

“Introducing a new process, let alone a new culture, to a large company takes a good strategy, executive buy in, strong evangelism, easy adoption patterns, and--most of all--patience. It might take years for a significant change like SLOs to become firmly established at a company. We’d like to emphasize that The Home Depot is a traditional enterprise; if we can introduce such a large change successfully, you can too. You also don't have to approach this task all at once. While we implemented SLOs piece by piece, developing a comprehensive evangelism strategy and clear incentive structure facilitated a quick transformation--we went from 0 to 800 SLO-supported services in less than a year.”

-- William Bonnell (Sr. Director, SRE -- The Home Depot)
You can do this.

Incremental progress buys time.
Any progress is time well spent.
Leave lots of on/off-ramps
It’s OK to stop for a while and come back when they’re ready.
Your teams can do this.
Other Googler talks

Today (Oct 2):

1:30-2:10pm in Beekman/Sutton North: **Jamie Wilkinson** on SLO Burn

2:25-3:05pm in Nassau: **Kristina Bennett** on Data Recoverability

Tomorrow (Oct 3):

9:25-9:45am in Grand Ballroom: **Jaana B. Dogan** on Tracing & Critical Path-Driven Development

1:30-2:10pm in Gramercy: **Seth Vargo** on Security Best Practices for Distributed Systems
Thanks! Q&A