When to NOT Set SLOs

Lots of strangers are running my software!

Marie Cosgrove-Davies
Product Manager, Google Cloud

mariecd@google.com
01 Background

a.k.a. Marie’s SRE Journey
Goals for Pivotal Web Services

Run a production system
Users rely on PWS to be available and responsive.

Enable eng org learning
PWS runs the newest Cloud Foundry code in the real world, often for the first time. This gives us opportunities to learn about scaling, operability, and user behavior.
How do we balance our goals?

If only a lot of very smart people had already thought about this...
Quick Vocab Review

Service Level Indicator ("SLI metric") - a **metric** that represents whether a specific user value is being delivered (i.e. “main page loads successfully within 500ms”)

Service Level Objective ("SLO threshold") - a **threshold** for a specific SLI metric that represents the level of reliability that we think our users want
Hey, it works!
02 Shocking twist!

As promised!
Where we started

This is confusing.

Many metrics, unclear meanings.

Difficult to know which thresholds are useful.

Hard to understand severity or urgency of problems.
Cool, so we give our customers our SLI metrics and SLO thresholds and we’re done, right?
The Value of SLO Thresholds

<table>
<thead>
<tr>
<th>Represent our user knowledge</th>
<th>Force us to think about user needs</th>
<th>Help us guide system architecture</th>
<th>Provide feedback on the way we run our systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>When do we think users will be dissatisfied at a level of service?</td>
<td>We can’t set an SLO threshold unless we have a working theory of what our users need from our product</td>
<td>We can avoid choosing insufficiently reliable or unnecessarily expensive systems</td>
<td>We trust our SLO thresholds to tell us if an issue is user-facing and needs immediate attention</td>
</tr>
</tbody>
</table>
Example 1: SLO threshold for *cf push*

99.5% SLO threshold - 3.6 hours downtime in 30 days

External users
Highly visible functionality
Many deploys / day

90% SLO threshold - 3 days downtime in 30 days

Internal team
No reputation risk
2 deploys / week
Example 2: SLO thresholds driving architecture (and saving money)

We have a vSphere cluster in a closet for internal use, but only have business hours response for it. Can we use it for this project?

If this database is unavailable, critical features don’t work. Do we need to make the database HA?

Does our product need to be multi-AZ? Multi-datacenter?
So what do we give to customers?

1. Standardized, easy-to-measure SLI metrics, ideally built into the product

2. Examples of SLO thresholds that have been chosen for different user profiles, and why

3. Guidelines on operational, architectural, or process practices that are needed to achieve specific SLO thresholds

4. (Bonus!) Program to guide customers through this process (for example, Customer Reliability Engineering)
This seems hard, can’t we just tell customers to use our SLO thresholds?

Sure, we can, we’re all adults, but if customer operations teams blindly adopt our SLO thresholds, it won’t be good for anyone.

- If an SLO threshold is too low or too high off the bat, you lose the trust of the operations team.

- If an SLO threshold is too low, the operations team loses the trust of the users. The users won’t think very highly of our product, either!

- If an SLO threshold is too high, costs will be higher than needed, and they’ll be less likely to buy more of your product.
To summarize...

1. If your customers run your software, provide them with SLI metrics.

2. SLO thresholds depend on user needs, so those vary by customer and instance.

3. The value of SLO thresholds comes as much from a team working through the process of setting them as from the numbers themselves. So help your customers find SLO thresholds that are right for them!
Questions?