When Production is Someone Else’s Black Box: Safely developing OSS for the enterprise

Isobel Redelmeier @ Open Source Leadership Summit
Hello there!

What’s the situation today?

Towards solutions

Q&A
Open Source Engineer @ LIGHTSTEP

Previously @ Pivotal FLUX

@1z03b1
Open Source Engineer @ LIGHTSTEP

Previously @ Pivotal, FLUX

@1z03b1

My accuracy SLA today is < 100%
What can open source learn from SaaS*?

*also PaaS, etc.
What can open source learn from SaaS*?

*also PaaS, etc.
<table>
<thead>
<tr>
<th></th>
<th>SaaS</th>
<th>OSS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Move fast</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Don’t break things</strong></td>
<td></td>
<td>(too much)</td>
</tr>
</tbody>
</table>
Move fast

Don’t break things (too much)
Move fast

Don’t break things (too much)

SaaS

OSS
Move fast

Don’t break things (too much)
Move fast

Don’t break things (too much)
Why even care about moving fast?
Why even care about moving fast?

- Faster feedback loops
Why even care about moving fast?

- Faster feedback loops
- De-risk risky bets
Why even care about moving fast?

- Faster feedback loops
- De-risk risky bets
- Deliver value sooner
Why even care about moving fast?

- Faster feedback loops
- De-risk risky bets
- Deliver value sooner
- Time is $$
Why even care about moving fast?

- Faster feedback loops
- De-risk risky bets
- Deliver value sooner
- Time is $$$
- Sometimes, we need to! 🦁
These are new challenges for open source
These are new challenges for open source
...or are they?
...or are they?
How often do you update your production…
How often do you update your production…

» Database?
How often do you update your production...

▪ Database?

▪ Operating system?
How often do you update your production...

- Database?
- Operating system?
- Other dependencies?
What happens when you do?
If SaaS gets continuous deployment…
If SaaS gets continuous *deployment*… can OSS get continuous *releasing*?
Continuous releasing:

- Would mean that ~every commit is released… or at least releasable
- Must be safe for end-users
- Must be easy for operators
• **SLA**: Service level agreement
• **SLA:** Service level agreement

• **SLO:** Service level objective
- **SLA:** Service level *agreement*
- **SLO:** Service level *objective*
- **SLI:** Service level *indicator*
These are in the control of operators
These are in the control of operators but we can help them
Operators need to be empowered to own their own observability
Operators need to be empowered to own their own observability.
Start by testing two scenarios:

1. Something cheap

2. Something that reflects a real environment

- Make performance tests easy to repeat
- Bake in observability instrumentation
We need to able to address tech debt
- Versions
- API changes and deprecations
- Migrations
Versions should be...

- Predictable
- Meaningful
- Simple
API deprecations start with API minimizations
Migrate schemas as necessary
Migrate schemas as necessary within your error budget
Migrate schemas as necessary within your error budget or in multiple steps.
Could we have...
Could we have...

• Automated updates?
Could we have...

- Automated updates?
- Phased updates?
Could we have...

- Automated updates?
- Phased updates?
- Shared telemetry?
Thank you!

@1z0b31 | isobel@lightstep.com