Put Some SRE in Your Shipped Software

Hard-won lessons from the world of SRE
The nature of the problem:

Software Sucks.

Once you’ve run software at scale, you have a deep understanding of how it is all tied together with loose string and hope.

We spend massive effort to operationalize our stacks.
The burning question:

Why Ship Software?

The trends are there. You can choose to see them or not.
Rules.

1. Crash landings should be both fast and controlled.
2. Post-mortems are fundamental.
3. Use circuit breakers.
4. Behavior is complex. Understand it.
5. Have a failure budget.
6. Instrumentation & observability have no equals.
Build upon the right layers

Crash Analysis

If you don’t know why it failed, you don’t know anything at all.
When in doubt or even curious

Expose Telemetry

Ideally, any question you would ask of a production system can be done so nondisruptively.
### Job Queues

<table>
<thead>
<tr>
<th>Queue</th>
<th>Running</th>
<th>Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_key_read</td>
<td>0.026ms</td>
<td>0</td>
</tr>
<tr>
<td>data_read</td>
<td>0.018ms</td>
<td>0</td>
</tr>
<tr>
<td>data_write</td>
<td>0.030ms</td>
<td>0</td>
</tr>
<tr>
<td>default_queue</td>
<td>0.011ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-2539d01-b137-3472b-8797-91f9d4f1dbd8</td>
<td>0.026ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-3cc4205af-40e7-4863-87e0-56b90230f6998</td>
<td>0.018ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-4bb9202b-3a46-4708-8ac1-eb5723fccc17</td>
<td>0.030ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-66a80c34-cf65-4825-0cc4-eff981ecdaa1</td>
<td>0.011ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-a212a487-25a5-47bd-8751-6330a4011677</td>
<td>0.044ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-a555b0-b9bd-431b-8662-9225ff2e733</td>
<td>0.021ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-a5ee7c74-1147-4afa-a665-1051ae497aba</td>
<td>0.019ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-ba5481cc-c3ac-4ff8-b5d3-e82926768bc40</td>
<td>0.028ms</td>
<td>0</td>
</tr>
<tr>
<td>journal-c482c214-90c9-4c9c-5f0c-8f4b45efdb06</td>
<td>0.020ms</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Latency of data_read latency

- p(50) = 1.3m-2.9m
- p(75) = 4.0m-5.0m
Known unknowns

Events & Distributed Tracing

A clearer story of what just happened.
During failure reconstruction, logs hold truth

**Logging for humans**

Computers talking to computers have better ways than logs. Logs are for computers talking to humans.
Real unknown unknowns are solved by:

**Dynamic Tracing**

bpftrace / eBPF DTrace
Your m8g, only need to be accessible

**Internalized MVP**

No additional apparatus. No additional deployment constraints.
Shipping software means more operators

Codify Operational Assessment & Procedures

More operators, less average knowledge.
Ensure procedures are repeatable.
Tools -> Solutions
Every effort to bring SRE techniques to software engineering makes SRE more accessible and useful in Cloud/SaaS engineering.
Thank You.

Theo Schlossnagle
CEO, Circonus

@postwait