Site Reliability Evangelism

Practice start-up within an established web-presence
A bit about us

Trade Me is NZ’s largest online auction and classifieds website

**Piers** (SRE Practice Lead) and **Catherine** (Technical Product Owner)
Structure & History

- 300 engineering staff
- Cloud Migration(s) ~18 months ago
- 3 Business Units
Dickerson’s Hierarchy of Service Reliability

- Monitoring
- Incident Response
- Postmortem / Root Cause Analysis
- Testing + Release procedures
- Capacity Planning
- Development
- Product
Monitoring

Quality is a shared responsibility

- The squad shares responsibility for releasing their code
- Engineers are invested in tools that can give them more confidence that their deploy was successful
- Main code contributor is involved from beginning to end of the cycle of the case so quality is considered throughout
- We don’t have a BAU team so engineers are responsible maintaining their systems
Monitoring

How we were monitoring our systems

- Synthetic production tests
- Deploy dashboards
- Golden Signal dashboards
Dickerson’s Hierarchy of Service Reliability

- Product
- Development
- Capacity Planning
- Testing + Release procedures
- Postmortem / Root Cause Analysis
- Incident Response
- Monitoring
Incident Review

We had

- a very structured process
- a Trade Me built bot - “Bergerac”
- a lot of active incidents
- but also some avoidance
Incident Review

The biggest reasons for delays in resolving and/or closing an incident are:

- Writing the POM and documenting all the actions.
- The delay chain of incidents and the feeling that everyone is too busy/resistant to perform actions and that if they go unread it's a wasted effort.
- People not wanting to write a POM.
- Waiting for POMs to be written.
- Getting POMs done AND reviewed to be signed off.
- From my limited experience it seems to be writing up the POM and getting everyone together for the post-mortem meeting.

From what I've seen people dread taking the time to fill out a potentially massive Post Mortem document and having a meeting after the fix. I guess people just want to get on with their work - maybe they're also fearing being blamed, are yet to see the value of it, or have had bad experiences? I'm just guessing.
Incident Review

This chart shows the number of issues created vs. the number of issues resolved in the last 300 days.
Incident Review - Changes Made

- Reviews expected on Sev1+ Sev2 incidents only
- Dropped Sev4’s completely
- Started a community of debrief facilitators
- SaaS incident management tooling with big benefits
- Learning as primary objective
  - Laundry lists of actions discouraged
  - Challenge Dashboard proliferation
  - Prefer transformative change
This time last year...
Incident Response - On call
Incident Response - On call
Cultural Reinforcement

The technical challenge of standing up on-call was just the first step

- Needed to work with our People Leaders to reinforce key ideas
- We have a great DELIVERY culture at Trade Me
- We need to turn “Build Teams” into “Build & Run Teams”
Service Level Objectives

We have an annual OKR to get SLOs wrapped around our core systems

- Struggled to get interest even when the Dev benefits were called out
- Prioritisation was hard (both order of approach, and getting time)
- Leveraged DR plans to source the true core systems
- Proper Work requests into our quarterly planning
The “Effort Pie”

Stream aligned teams

- Operation: 20%
- Personal development: 10%
- Feature development: 50%

Platform teams

- Operation: 20%
- Personal development: 10%
- Feature development: 50%
Operational Review Meetings

- SLOs / Key Performance Stats
- Any incidents, review or actions since the last meeting
- Any emerging risks and / or related actions that need to go into the risk register
- Maintenance plans - what to spend the 20% time on next quarter

<table>
<thead>
<tr>
<th>SLO</th>
<th>Last 2 hours...</th>
<th>Last 24 hours...</th>
<th>Last 7 days...</th>
<th>Last 28 days...</th>
</tr>
</thead>
<tbody>
<tr>
<td>95%</td>
<td>94.91 %</td>
<td>93.6 %</td>
<td>93.86 %</td>
<td>94.08 %</td>
</tr>
<tr>
<td>1 day</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99.9%</td>
<td>100 %</td>
<td>100 %</td>
<td>100 %</td>
<td>100 %</td>
</tr>
<tr>
<td>1 day</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95%</td>
<td>96.68 %</td>
<td>96.27 %</td>
<td>96.12 %</td>
<td>96.19 %</td>
</tr>
<tr>
<td>1 day</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In Summary

- Choose a safe, foundational project to build early credibility
- Simplify the messages / over-communicate
- Investing in cultural change is a long game
- SRE is not just for Site Reliability Engineers
- Get reliability work onto committed roadmaps
Referenced Resources

- John Tucker’s System ERD diagram
  https://john-tucker.medium.com/backstage-by-example-part-1-a18e74849240
- Ask Ms O11y blog - ...am I a monster?
  https://www.honeycomb.io/blog/devops-on-call
- Team Topologies by Manuel Pais and Matthew Skelton
- Site Reliability Engineering by Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy