LESSONS FROM AN ASSET MANAGER'S FIRST EMBEDDED SRE

Callum Donald

AGENDA

Setting Expectations	What is Embedded SRE going to do?
Building Trust	Before you focus on building alerts
Maximising Returns	From investments in reliability you have already made
Investing Earlier	Create new reliability investment opportunities

SETTING EXPECTATIONS

Embedded SRE *noun*

An engineer with reliability expertise, embedded within a product or platform team, shaping culture, process and tooling to improve reliability.

Not an auditor.

Not a dedicated pager-holder.

A partner in reliability.

SETTING EXPECTATIONS

50
systems

12
months

1
engineer

Setting Expectations

Scope

Six Critical Services | Trusted Leaders

Measurable

Known Themes | Telemetry | Capacity | Resiliency Risks

Mutual Commitment

Visibility | Training | Prioritisation | Insurance

SETTING EXPECTATIONS

Experiment

Look for inspiration externally. Learn from industry.

Adapt

Become comfortable adapting or abandoning best practice.

Reality

There is no one size fits all for SRE. Work within your operational reality.

BUILD TRUST

Before you build alerts

BUILD TRUST

Alert Fatigue

“ a phenomenon in which **repeated exposure** to frequent or non-actionable alarms leads to sensory **overload,** emotional **strain,** and a gradual **desensitization** or reduced responsiveness … increasing the risk of delayed or **inadequate alarm responses**”

– Michels, Elizabeth Anna Mathilde et al. “Alarm fatigue in healthcare: a scoping review of definitions, influencing factors, and mitigation strategies.” *[1]*

[1] https://pmc.ncbi.nlm.nih.gov/articles/PMC12181921/

BUILD TRUST

Inventory

Signal	Intention	Urgency	Decision
Database Error	We need to know when the tempdb fills	Low	Keep / fix / **remove**

BUILD TRUST

Symptomatic Alerting

Root causes are infinite and relatively unpredictable

Symptoms are finite and guarantee impact

BUILD TRUST

Symptomatic Alerting

- Four Golden Signals
  - Latency
  - Traffic
  - Errors
  - Saturation
- R.E.D
  - Requests
  - Errors
  - Duration

BUILD TRUST

Symptomatic Alerting

> “Does this alert definitely indicate that users are being negatively affected?”

Chapter 6 – Monitoring Distributed Systems,
Google SRE Book [1]

[1] https://sre.google/sre-book/monitoring-distributed-systems/

MAXIMISE RETURNS

From investments you have already made

MAXIMISE RETURNS

Retrospective Template

1

Timeline

”Why didn’t we notice that earlier”

Opportunities to reduce your TTX ...

- Acknowledge
- Detect
- Resolve

2

Proximate Trigger

The closest cause before the incident

Ask the ‘Five Whys’

- API was slow ... why?
- Database was slow ... why?
- Query plan was inefficient ... why?

3

Contributing Factors

Incidents don’t happen in isolation

What made the incident possible

- Pre-existing conditions
- Contributing failures or factors
- Environmental factors

MAXIMISE RETURNS

Retrospective Template

4

Feedback loops

How was the system controlled?

Consider both humans and systems

- What signals were produced?
- Who received them?
- What actions did they take?

5

Resolution

How was the incident resolved?

What was the key fix or mitigation?

- What was done in the short-term?
- Are there long-term mitigants?
- What risks remain open?

6

Lessons Learned

What are the key takeaways?

Are they applicable across the org?

- What went well (or not)
- Assumptions Challenged
- What can we change?

MAXIMISE RETURNS

Retrospective Template

7
Action Items

What actions can limit recurrence?

Think tactically and strategically

- Clear
- Measurable
- Assigned

Over Three Years ...

100 Retrospectives Run

3000 Retrospective Views

30% Annual Incident Reduction

INVEST EARLIER

Consider supportability earlier in the product development lifecycle

INVEST EARLIER

Supportability *noun*

Non-functional requirement describing the ability to:

Know **when** something is broken

Quickly identify **where** or **why** it is broken

Take quick and effective **action**

INVEST EARLIER

Healthy disregard for boundaries

Everyone's responsibility

Use existing processes

IN SUMMARY

Start with trust

Inventory Alerts. Classify them, be ruthless about the noise.

Run one retrospective

Focus on the discussion. Ask '*why?*'

Ask the supportability questions early

How do we know if and where its broken? What controls do we need?