ELSEVIER

SRE in the Enterprise

Year 0

About Me

- DevOps Director in Elsevier Research Technology
- In tech since 2000
- I work to bring DevOps goodness to our engineering squads

About Elsevier

Founded in 1880, as a publishing company

Part of the RELX Group – a FTSE 100 business with a market cap of £40b

8000 Employees – our parent company has 33,000

- World’s largest platform for peer-to-peer scientific research
- 1700 Technical staff
- 17m scientific articles
- 40000 e-books
- 3500 journals

Our Technology Evolution

Print Publishing
- Books
- Journals

Digitisation & Online CD’s
- Static websites

Cloud
- Lift & shift
- Re-writes

Agile & DevOps
- Increased velocity & quality
- Autonomous squads

ELSEVIER

Challenges – Legacy & Market

- 140 years old - we have many more years as a print publisher than as a tech company
- Lots of products have been around for decades
- Lots of acquisitions means different cultures, and different ways of working
- We are a blue chip, dividend paying company with commitments to the City & pension funds
  - It’s difficult for us to “Move Fast & Break Things”
  - Technical evolution tends to be in phases
- The market is changing quickly

Challenges – Reliability

- We have dozens of products & services
    - Different definitions & expectations of availability
- Increased interdependence of services
    - An issue in one system affects many others
- Real-time, life-dependent data as we move more into healthcare.
    - System reliability is increasingly important

ELSEVIER

Reliability

our next phase

What is SRE?

It’s what happens when you ask a software engineer to design an operations function.

Using automation to improve reliability

Belief that a prerequisite to success is availability.

SRE classifies “Reliability as a Feature”

ELSEVIER

Principles of SRE

1. Operations is a software problem
2. Manage by Service Level Objectives
3. Work to minimise manual, repetitive work
4. Automate this year’s work away
5. Move fast by reducing the cost of failure – think fast rollbacks, microservices
6. Share ownership with developers
7. Use the same tooling, regardless of function or job title

Practices of SRE

In general, an SRE team or person, is responsible for availability, latency, performance, efficiency, change management, monitoring, emergency response, and capacity planning.

There is an agreement, with Product teams, to stop releases if reliability targets are compromised

SRE’s should be top-level engineers who understand both Operations and Software Engineering

Terminology

SLA – Service Level Agreement

- A formal agreement with your customers with regard to service availability.

SLO - Service Level Objective

- Usually an internal metric, used to inform the SLA
- Should be more stringent than the SLA. Ie 99.99% vs 99.9%

SLI – Service Level Indicator

- The signals used to inform your SLO (monitoring)

Terminology

Four Golden Signals

- The most fundamental metrics to measure
  - Latency – the time it takes to service a request
  - Traffic – how much demand is being placed on your system ie. Requests per sec
  - Errors – the rate or requests that fail. Ie. 500’s, etc
  - Saturation – how “full” your system is ie. i/o, memory, etc

Terminology

Error Budget

- The difference between actual uptime, and the SLO, is the error budget.
- As long as there is error budget remaining—new releases can be pushed.
- The idea is that no system can or should be 100% available.

So what’s the plan?

- Three pilot projects
- 6 month-*ish* duration

Why them?

- Existing reliability issues
- Enthusiastic leadership

So what’s the plan?

- We’ll start by:
  - developing an SLO (and building out the necessary SLI’s)
  - Establishing a strong incident management procedure
  - Instituting blameless post-mortems
  - Establish an SRE guild so we can help each other out
- Each pilot will have a dedicated SRE for the duration
  - Champions SRE
  - Helps establish SLO’s, SLI’s, processes
  - Trains the rest of the squad
  - Can be either a DevOps/Infra Engineer or Software Engineer

And after that?

- We iterate and improve as we expand to different products and services.
- Figure out what a good resourcing model looks like. Ie Shared SRE team, dedicated SRE’s per squad, consultation model, etc.
- Hire or re-train some full-time SRE’s

- We will learn, make mistakes, and improve ourselves.

What does success look like?

A few of the things we are hoping to see:

- A common language that bridges teams and BU’s
- More collaboration between teams
- Improved MTTR
- Improved velocity

- We believe SRE will improve service for our users.
- Happy users = better research for the world

- We believe it will improve the lives of our Engineers
- Happy Engineers = Great software for our users

Experimentation and Innovation

- Innovation can and does apply to “traditional” businesses with market constraints.
- Build a culture of experimentation.
- SRE (and DevOps) is for every everyone, not just cool tech companies.

Thank You

ELSEVIER