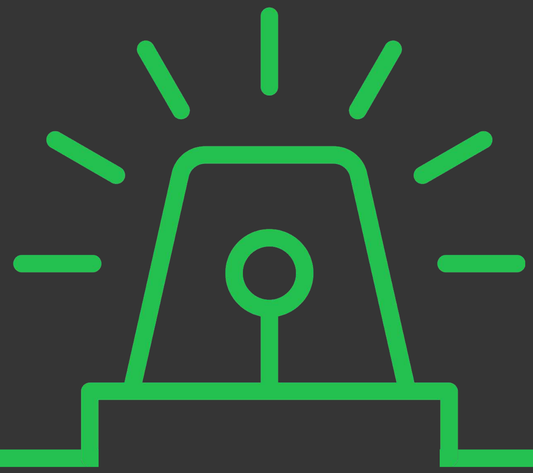
The Lifecycle of a Service

Matt Stratton
DevOps Advocate & Thought Validator, PagerDuty




IMG
IMG
18px
image:

Service Ownership means people take responsibility for what they deliver, at every stage of a service's lifetime.

Communicate across your organization with partners and stakeholders

What is a service?

A service can be a lot of things

Microservice

Slice of a monolith

Internal tool

Piece of functionality

Component

Shared infrastructure

Feature

A service can be a lot of things

If it provides value to other people, it’s a service

Define what a “service” means to you

A service is a discrete piece of functionality that provides value that is wholly owned by a team

Shared understanding

Who is responsible?

“Service mitosis”

Service definitions help with problem resolution

What about a monolith?

Roles in service ownership

Development Team

Your service should make sense to other people who will interact with it

Naming

Be specific

Names that are specific

- “User authenticator”
- “Payment processor”
- “Shopping cart”
- “Login”
- “Report generator”
- “Email tracking code”

Less amazing names

- PacMan (unless you’re actually building PAC-MAN, which I doubt)
- Apollo
- BurgunDB
- Artemis

Descriptions

- What is the intent of this service, component, this slice of functionality?
- How does this thing deliver value?
- What does it contribute to?
- How will this impact customers?

Dependencies

- Look for circular dependencies
- Is there a single point of failure?
- Who consumes this service?
- What does it depend on?

API

- Versioning
- Clear documentation / examples

Tiers of services

Tier 1 Services at PagerDuty

- 24/7 on-call
- Multiple levels of robustness
- Disaster recovery plan
- Clear and updated runbook

Tier 2 & 3 Services at PagerDuty

- Monday-Friday support expectation
- Supporting functionality, not critical path
- New services that are not generally available

Sustainability team

Runbooks

Alerting

Robustness and reliability

Program management

Responsibilities of program management

- Defining what ‘done’ is
- Emotional awareness of stress of the rest of the team
- Connective tissue work between different teams and features (help understand and mitigate dependencies)
- Awareness of what it means to pull people away from other projects to solve a problem

Product owner

Customers are always asking for uptime, performance, and security – they just don’t usually use those words

Management

- Make room in the roadmap for investing in tech debt
- Encourage a culture of cooperation and sharing
- Set goals that balance business priorities with achievable engineering goals

Going deeper

What are you observing about this service?

Observability vs monitoring

Baron Schwartz

Founder and CTO, VividCortex

Monitoring tells you whether the system works. Observability lets you ask why it's not working.

Empathy-driven alerting

A brief overview of SLA/SLO/SLI

Service Level Indicators (SLI)

- Latency
- Throughput
- Availability

Service Level Objectives

- Made up of SLI’s
- Measured over time
- Not contractually set

Service Level Agreements

- Composed of SLO’s
- Contractually/legally binding
- Basically, this is where you owe your customer money

The “hadness” point

Alert on SLO’s

How does a team respond to this service?

Escalation policies

DevOps Model

First level

Second level

Third level

Escalating

Manual escalations

Other escalation models

- Central Ops
- Hybrid Ops

Tuning your service

Investigate patterns

What alerts do you *actually* need?

Suppression of non-actionable alerts

Understand business impact

Lifecycle steps

Designing a new service

- Understand the customers (product is a key role here)
- Load testing / staging
- Ensure SRE / sustainability teams are involved early
- Define SLI/SLO/SLA
- Identify alerting requirements
- Documentation (API, runbook, functional service registry if applicable)
- Perform all security checks

Maintaining and iterating

- Version the service API
- Communicate to consumers
- Proactive maintenance
- Address tech debt consistently
- Testing and deploying/releasing the service (CI/CD, testing in prod, etc)

Retiring a service

- Identify consumers
- Determine business impact of retiring
- Communicate / offboard consumers

Service ownership includes communication, compromise, and commitment.

Acknowledgements

Lilia Gutnik - @superlilia
Julian Dunn - @julian_dunn
Charity Majors - @mipsytipsy
Baron Schwartz - @xaprb

Images provided by

pixabay

If you enjoyed this talk, here's more about me

arresteddevops.com

devopsdayschi.org

twitter.com/mattstratton

speaking.mattstratton.com