September 27, 2019

Postmortems or How We Learn from Failures

Building an incident review culture inside a large fintech

What we’ll cover

Agenda Plan

- Why we need postmortems
- How Etsy and Google do it
- Our template and process
- Real failures and takeaways

Part 1

01

Why We Need Postmortems

When teams grow, failures become cross-team

Evolution 2016 → today

Teams grow — failures get harder

From simple single-team incidents to cross-team failures

Late 2016

3 teams, simple within-team fails

Growth

more teams — cross-team fails

Today

cross-department incidents

What process must produce

Jobs of a postmortem

- Capture the incident
- Quantify business/user impact
- Describe firefighting
- Actions + owners prevent recurrence

Goals What we wanted

Part 2

02

How Others Work with Postmortems

Etsy and Google as key references

Where we learned

Sources References

Blameless postmortem sources

- Etsy — Just Culture
- Google SRE Book and SRE Workbook
- Google’s public postmortem template
- Etsy — Debriefing Facilitation Guide

Etsy: review without blame

Etsy Blameless

Blamelessness gets honest data

- Goal — find facts
- No blame, more detail
- Fear breeds CYA engineering
- Just Culture: error ≠ at-risk behavior

Postmortem = learning artifact

Google SRE Book: review, not punishment

- Blameless and constructive
- Every postmortem is reviewed
- No review — better not to write
- Action items: concrete and owned

Google SRE Book

A good postmortem vs a bad one

Google SRE Workbook

The Workbook breaks down both with examples

× **Bad**

- Vague summary
- No concrete actions
- Hunting for a culprit

✓ **Good**

- Clear timeline
- Action items with owners
- Blameless tone

Criteria: clarity, concreteness, blamelessness

Part 3

03

Our Postmortem Template

What we borrowed and how we adapted it

Template Structure

Five sections cover every question

Equally easy to write and to read

- Issue Summary — one sentence
- Timeline — what and when
- Root Cause — technical and systemic
- Resolution & Corrective — with owners

Part 4

04

Failures of “a friend”

Real cases and what we took from them

Case 1 SEO

Meta tags fell off in a release

A shared component with no owner

- Front-end release — SEO drop
- Tightly coupled monolith
- 20+ teams edit one library
- No autotests, no regression check

Takeaways: meta-tag checks in regression, rework the shared component

Case 2 Kubernetes

Probes are requests too

Django, long requests and probes

- Single-threaded Django
- Some API requests are long
- Probes queued with user requests
- Traffic spike — restart loop

Takeaways: know your runtime, design for degradation

Case 3 Cache

Cache needs versioning too

Everyone versions the DB, few version the cache

- Bumped the schema version
- Changed the expected response
- Stale cache returned — 404s for an hour
- Stage had no traffic mirroring

Takeaways: mirror traffic to stage, set cache versioning rules

What matters

Summary Key points

Rules that work for us

- Blameless — or truth disappears
- Template required — or not comparable
- Action items without owners don’t get done
- The review is about the system, not people

- a postmortem without action items is just a report

polomodov.tech

Thank you!

All slides and links are in the Telegram channel

Alexander Polomodov

Technical Director & Fellow,
T-Technologies



@book_cube