Post-Mortem Template

Post-Mortem

> **Project:** Bilko **Version:** 0.1 **Date:** 2026-02-23 **Author:** Ops Architect **Status:** Draft (Template — fill in per P0 incident) **Reviewers:** Tech Lead, Alem Bašić

Document History

Version	Date	Author	Changes
0.1	2026-02-23	Ops Architect	Initial draft

INSTRUCTIONS

Post-mortems are required for all P0 incidents and recommended for P1. Schedule within 5 business days of incident resolution.

Blameless culture: This document is about systems and processes, not people. The goal is to learn and prevent recurrence, not to assign blame.

Create a new file: `POST-MORTEM-YYYY-MM-DD-<title>.md` in `docs/operations/post-mortems/`

Post-Mortem: [INCIDENT TITLE]

Post-Mortem Date: YYYY-MM-DD **Incident Date:** YYYY-MM-DD **Incident Reference:** INC-YYYY-MM-DD-NNN **Facilitator:** [Name] **Attendees:** [Names] **Duration of post-mortem session:** [X minutes]

---

1. Executive Summary

What happened: [2-3 sentences: the incident, impact, and resolution]

Why it happened: [1-2 sentences: root cause in plain language]

What we're doing to prevent recurrence: [1-2 sentences: top action items]

---

2. Impact Summary

Metric	Value
Incident duration	X hours Y minutes
Detection time	X minutes from first symptom
Response time	X minutes from alert to first action
Users impacted	[All / Specific org / None]
Financial records affected	[None / Describe]
Downtime cost (est.)	[€X in lost productivity / TBD]
GDPR breach notification required	[Yes / No]

---

3. Timeline (Detailed)

Time (CET)	Event	Who	Notes
HH:MM	[Event]	[Person]	[Notes]

Key timestamps:

- **First symptom:** HH:MM
- **Alert fired:** HH:MM (detection lag: X min)
- **Incident declared:** HH:MM (response lag: X min)
- **Root cause identified:** HH:MM (diagnosis duration: X min)
- **Fix applied:** HH:MM
- **Service restored:** HH:MM
- **Incident closed:** HH:MM
- **Total user impact duration:** X min

---

4. Root Cause Analysis

What happened technically

[Detailed technical explanation of the failure chain. Be specific: which component, which code path, which query.]

For Bilko financial incidents, this section must include:

- Which accounting module was affected (VAT / double-entry / invoice calc / currency)
- Was any financial data corrupted? If yes, which organizations, which time window
- Were NUMERIC(19,4) values preserved correctly during the incident?

Why it happened

[The "5 Whys" — trace back to the systemic cause]

1. **Why** did users lose access? → API returned 503 errors
2. **Why** did the API return 503? → Railway service restarted due to OOM
3. **Why** did the service run out of memory? → Invoice PDF generation loaded entire result set into memory
4. **Why** did we not catch this? → No memory profiling in development, load testing not done
5. **Why** was there no load testing? → No performance test plan existed

Root cause (systemic): [e.g., "Lack of memory usage monitoring and load testing prior to feature launch"]

Contributing Factors

Factor	Category	Severity
[Factor]	[Process / Code / Infrastructure / Communication]	[High / Med / Low]

---

5. What Went Well

- [e.g., BetterStack alert fired within 2 minutes of downtime starting]
- [e.g., Rollback procedure was documented and worked on first try]
- [e.g., Financial data integrity was preserved — no accounting records corrupted]
- [e.g., User communication was clear and timely]

6. What Went Poorly

- [e.g., No memory usage alerting was configured before launch]
- [e.g., The runbook did not cover OOM scenarios]
- [e.g., Detection took 8 minutes because uptime check interval was 5 min]
- [e.g., Only one person knew how to access Railway logs]

---

7. Action Items

High Priority (P0/P1 — complete within 2 weeks)

#	Action	Category	Owner	Due	Status
1	[Action]	Prevention	[Name]	YYYY-MM-DD	Open
2	[Action]	Detection	[Name]	YYYY-MM-DD	Open

Medium Priority (P2 — complete within 1 month)

#	Action	Category	Owner	Due	Status
3	[Action]	Process	[Name]	YYYY-MM-DD	Open

Low Priority (P3 — add to backlog)

#	Action	Category	Owner	Due	Status
4	[Action]	Nice-to-have	[Name]	Backlog	Open

Action categories: Prevention, Detection, Response, Documentation, Process, Tooling

---

8. Lessons Learned

Technical Lessons

[What did we learn about the technology, the system design, or the code?]

For Bilko financial system incidents:

- [e.g., NUMERIC(19,4) Decimal arithmetic must be tested under concurrent load]
- [e.g., VAT calculation should be validated server-side even if client sends computed totals]

Process Lessons

[What did we learn about our operations process, monitoring, or communication?]

Culture Lessons

[What did we learn about team practices, communication patterns, or organizational factors?]

---

9. Metrics Targets for Next Quarter

Based on this incident, these metrics are now tracked:

Metric	Current	Target	By
Mean time to detect (MTTD)	X min	< 3 min	YYYY-MM-DD
Mean time to respond (MTTR)	X min	< 10 min	YYYY-MM-DD
Mean time to resolve (MTTR)	X min	< 60 min	YYYY-MM-DD

---

10. Follow-Up Schedule

- ☐ Action items tracked in GitHub Issues (label: `post-mortem-action`)
- ☐ 2-week check-in: verify P0/P1 actions completed
- ☐ 1-month check-in: verify P2 actions completed
- ☐ Next post-mortem: review if similar incidents recurred

Approval

Role	Name	Date	Signature
Facilitator			

Role	Name	Date	Signature
Reviewer	Alem Bašić		

Revision #3
Created 2026-02-24 23:11:22 UTC by John
Updated 2026-05-31 20:04:09 UTC by John