Blameless Incident Post-Mortem Template

Incident ID: _______________

Date of Incident: _______________

Post-Mortem Date: _______________

Facilitator: _______________

Attendees: _______________

Incident Summary

In 1-2 sentences, what happened?

Example: "On 2026-01-15 at 14:32 UTC, the API gateway became unresponsive causing a 27-minute outage affecting 45% of users."

Severity & Impact

Severity Level: P0 (Critical) | P1 (High) | P2 (Medium) | P3 (Low)

Impact:

- **Users Affected:** [number or percentage]
- **Duration:** [start time - end time, total minutes]
- **Business Impact:** [revenue, reputation, SLA breach]
- **Systems Affected:** [list of services]

Timeline

Time (UTC)	Event	Actions Taken
14:32	Alert fired: API gateway CPU at 100%	On-call engineer paged
14:35	Confirmed widespread user reports	Incident declared, war room started
14:40	Identified memory leak in v2.3.1 deployment	Began rollback procedure
14:52	Rollback to v2.3.0 completed	Monitoring recovery
14:59	Service fully restored	Incident resolved
15:15	Post-incident monitoring complete	All systems normal

Root Cause

Technical Root Cause:
What was the immediate technical cause?

Example: "A memory leak introduced in release v2.3.1 caused the API gateway to exhaust available memory, leading to process crashes and service unavailability."

Contributing Factors:
What systemic issues made this possible?

1. Load testing didn't include 24-hour soak tests to detect memory leaks
2. Memory usage alerts set too high (90% threshold)
3. Gradual rollout skipped due to release pressure
4. Memory profiling not part of standard review process

Detection

How was the incident detected?

- □ Automated monitoring alert
- □ Customer report
- □ Internal team member
- □ Other: _______________

Time to detection: _______ (from incident start to first awareness)

Could detection be improved? [Yes/No] - If yes, how?

Response

What Went Well ☑

1. On-call engineer responded within 3 minutes of page
2. War room assembled quickly with right stakeholders
3. Rollback procedure executed smoothly
4. Communication to customers timely and transparent
5. Team remained calm and followed runbooks

What Went Wrong ✗

1. Root cause took 20 minutes to identify (memory leak not immediately obvious)
2. Monitoring dashboards didn't show memory trends clearly
3. Rollback took longer than target (12 min vs 5 min SLA)
4. Some customer-facing teams weren't notified promptly
5. No immediate way to failover to backup region

Action Items

Format: [Priority] Action - Owner - Due Date - Status

Immediate (This Week)

- [P0] Add memory trend monitoring to primary dashboard - DevOps Team - 2026-01-18 - Open
- [P0] Implement 24-hour soak tests in CI/CD - QA Lead - 2026-01-20 - Open
- [P1] Lower memory alert threshold to 75% - SRE Team - 2026-01-17 - Open

Short-term (This Month)

- [P1] Improve runbook with memory leak troubleshooting steps - Tech Lead - 2026-02-01 - Open
- [P1] Optimize rollback automation to hit 5-min SLA - Platform Team - 2026-02-15 - Open
- [P2] Add customer success team to incident notification workflow - Operations - 2026-01-30 - Open

Long-term (This Quarter)

- [P2] Implement multi-region failover capability - Architecture Team - 2026-03-31 - Open
- [P2] Adopt memory profiling in code review process - Engineering - 2026-02-28 - Open
- [P3] Conduct chaos engineering exercise - SRE Team - 2026-03-15 - Open

---

Lessons Learned

Technical

- Memory leaks in long-running services require soak testing, not just load testing
- Memory monitoring needs better visualization of trends over time
- Automated rollbacks need optimization to meet SLA targets

Process

- Release pressure shouldn't skip gradual rollout procedures
- War room communication worked well, extend to all stakeholder teams
- Runbooks need regular testing and updates

Cultural

- Team handled pressure well with blameless mindset
- Cross-functional collaboration was strong
- Post-mortems are learning opportunities, not blame sessions

---

Follow-up

Action Item Review Date: ________________

Post-Mortem Published: ☐ Internal Wiki [ ] Public Status Page [ ] Team Retrospective

Responsible for Follow-up: ________________

Blameless Culture Reminders

☑ **Do:**

- Focus on systems and processes, not people
- Ask "what" and "how", not "who" and "why"
- Assume everyone acted with best intentions given their information
- Celebrate what went well
- Learn and improve

✗ **Don't:**

- Assign blame to individuals
- Focus on punishment
- Assume malice or incompetence
- Skip follow-up on action items
- Hide incidents from the team

Remember: The goal is learning and improvement, not finger-pointing. Every incident is an opportunity to make our systems more resilient.