

Incident Response Phrases

English phrases for incident response — acknowledging alerts, status updates, escalation, and blameless postmortems.

<https://coderslingo.com/resources/cheatsheets/incident-response/>

Acknowledging the alert

When a page fires, respond fast so the rest of the team knows it's covered.

- **I'm on it.**
The fastest possible acknowledgement.
- **Acknowledged — investigating now.**
Confirms you own it and have started.
- **Picking this up — looking at the dashboards.**
Says what you're doing first.
- **I can take point on this one.**
"Take point" = lead the response.
- **Paging in — give me a minute to get context.**
Honest when you're just joining.

Status updates

Post short, regular updates even when there's no progress. Silence makes people anxious.

- **Current status: error rate is elevated on the checkout service, cause unknown.**
State the symptom and what you do/don't know.
- **Next update in 15 minutes.**
Set expectations so others stop asking.
- **No change yet — still digging into the logs.**
"No change" is a valid, reassuring update.
- **We've identified the likely cause: a bad config push.**
"Likely cause" — don't over-claim certainty.
- **Mitigation in progress — rolling back the last deploy.**
Distinguish mitigation from root-cause fix.
- **Error rate is trending down after the rollback.**
"Trending down" = improving, not yet resolved.
- **Customer impact: roughly 5% of checkout requests are failing.**
Quantify impact when you can.

Declaring severity & escalating

Name the severity clearly and pull in help early. Over-escalating is cheaper than under-escalating.

- **Declaring this a SEV-2 — major degradation, no full outage.**
State the level and a one-line justification.
- **This is a SEV-1, customer-facing outage. Spinning up an incident channel.**
Highest severity, full outage.
- **I'm escalating to the database team — this is beyond my area.**
Escalate by team or expertise.
- **Can we page the on-call for payments? We need eyes on this.**
"Need eyes on this" = need someone to look.
- **Looping in the incident commander now.**
"Looping in" = adding someone to the response.
- **Let's pull in the SRE on call before this gets worse.**
Pre-emptive escalation when risk is rising.

Resolution & all-clear

Close the loop clearly so everyone knows it's over and what comes next.

- **Mitigated — error rate is back to baseline.**
"Mitigated" = symptoms gone, root cause maybe not.
- **All clear. Service is fully recovered.**
The explicit end-of-incident signal.
- **Closing the incident. Postmortem to follow.**
Promise the write-up immediately.
- **Monitoring for another 30 minutes before we stand down.**
"Stand down" = end active response.
- **Thanks everyone for jumping on this so quickly.**
Acknowledge the team before closing.

Postmortem language (blameless)

Postmortems describe systems and decisions, never people. Keep the language neutral and factual.

- **This is a blameless postmortem — we're here to fix systems, not assign fault.**
Set the tone up front.
- **Contributing factors: a missing alert, an unreviewed config change, and a slow rollback.**
"Contributing factors", not "the person who..."
- **The timeline: alert fired at 14:02, mitigated at 14:31.**
Build a factual, timestamped timeline.
- **Root cause: a config change disabled retries without a guardrail.**
Describe the mechanism, not the author.
- **What went well: detection was fast and the rollback was clean.**
Always include what worked.
- **Action items: add a pre-deploy config check (owner: SRE, due next sprint).**
Each action item needs an owner and a date.
- **We were lucky here — let's remove the luck with an automated check.**
Turn a near-miss into a concrete improvement.

