What is on-call management? All you need to know

This guide explains on-call management from the ground up: schedules, rotations, five on-call models, and a 4-step setup that keeps the on-call load fair.

UPDATED OCTOBER 2026

TL;DR

On-call management decides who responds to incidents outside office hours and how the alert reaches them. A good setup covers five things: who's on call and when, how alerts escalate, which channels they arrive on, how shifts hand over, and how fairly the on-call load is shared. When it works well, every hour has an owner and a backup. The on-call load also stays light enough for people to get proper rest.

What is on-call management?

On-call management is the process that decides who responds to an incident at any hour and gets the alert to them. A team of five might take turns being on-call for a week each, with a second person as backup. When checkout fails at 2 AM, that week's person on-call gets a phone call and the backup is ready if they miss it.

The person on-call doesn't have to fix every problem alone. Their job is to acknowledge the alert, assess the severity, and bring in help when it's needed.

For every shift, the setup settles two things:

- Who carries the pager, and for how long
- Who the backup is if the first person doesn't respond

Why is on-call management important?

It matters because problems don't wait for office hours. Without it, an alert either reaches nobody or reaches everybody, and the fix starts late either way.

Say your food delivery app stops taking orders at 11 PM on a Saturday. With nobody named for the weekend, the alert lands in a group chat that everyone has muted. The team finds out on Monday, after a weekend of lost orders. With a weekend rotation in place, one engineer gets a phone call as soon as the incident triggers and has orders flowing again within the hour.

A good setup pays off in three ways. One named owner acknowledges faster than a busy group chat, which shortens mean time to acknowledge (MTTA). The fix starts minutes after the incident triggers instead of hours later, so downtime stays short. And fair rotations keep the load within Google SRE's limit of two incidents per 12-hour shift, so the team stays rested.

What are the core components of on-call management?

1. **Scheduling and rotations:** An on-call schedule sets which hours need cover, and a rotation decides who takes each turn. Rotations usually run in daily or weekly blocks.
2. **Escalation policies:** An escalation policy moves an unacknowledged alert to the next person after a set wait time. A common path runs from the primary to the secondary and then to the team lead.

3. **Multi-channel alerts:** Alerts that flow through more than one channel give the person on-call more chances to pick up an incident. An email or a Slack message is easy to miss at night. If nobody acknowledges it within 5 minutes, a phone call follows, so the incident still gets picked up.
4. **Handoffs:** A handoff passes context to the next person on-call: open incidents, systems acting strangely, and alerts that fired for no reason. A short note at shift change saves them from starting cold.
5. **Runbooks and documentation:** A runbook is a short guide for a common issue: what to check, how to fix it, and who to call. Up-to-date runbooks and a simple map of how the system fits together let the person on-call fix problems at 3 AM without waking the expert.
6. **Health tracking:** MTTA, escalation rate, and night pages per person show whether on-call is working and whether the load is fair. Spike's shift reports show who carried the pager and for how long.

Which on-call rotation model fits your team?

If your team has	The model that fits	Why it works
5 or more people in one time zone	Primary and secondary	Every shift has a backup, and newer engineers learn with help nearby
3 or 4 people with similar skills	Rotation-based	Simple turn-taking keeps the load even
Offices in 3 or more time zones	Follow the sun	Each region covers its own daytime, so nobody works nights
Part-time staff or fixed hours	Fixed schedule	Each person keeps the same slot every week
Specialists who own separate systems	Skills-based	Alerts go straight to the person who knows that system

Each model comes with a trade-off. Follow the sun depends on clear handover notes between regions. Primary and secondary keeps two people on-call at once, so each person spends more weeks on-call than in a simple rotation. Skills-based rotations can wear out the experts. Rotation-based and fixed schedules are the easiest to run, as long as the shift length suits the team.

Setups often mix models as they grow, and Spike's article on offbeat on-call rotations shows a few less common patterns that work.

How do you set up on-call in 4 steps?

Step 1: Pick the services that need 24/7 cover

A good starting question is "Which alert would I want to wake up for at 3 AM?" For an online store, that's payment processing and checkout. For a SaaS product, it's probably login and the core API. Services where downtime affects users or revenue are the ones to cover first.

Step 2: Choose who goes on-call

The people on-call need enough knowledge of the system to fix simple problems and to know when to ask for help. Team size matters here. Google's SRE book treats eight engineers as the minimum for a single-site team that runs a primary and a secondary, so each person is on-call one week in four.

Location matters too. A team in one city needs night shifts, while a team spread across time zones can share the day between regions. Smaller teams can still run on-call by covering only their most critical services at night. Before adding anyone to the rotation, it helps to have runbooks, clear escalation paths and tool access in place. Spike's post on who should be on-call explains why these matter more than seniority.

Step 3: Choose a rotation pattern

A weekly rotation is a good place to start for groups of five or more. Smaller groups might find a 2-day or 4-day rotation easier to sustain. Each shift works well with a secondary who gets the alert if the primary doesn't respond within 5 minutes.

Step 4: Connect your tools and set the escalation

Monitoring tools like Datadog or Prometheus are what spot a problem in the first place. On their own, they don't know who is on-call. Spike fills that gap as an incident management tool with on-call built in. Once your monitoring tools are linked to it, each alert goes straight to the person on shift. If that person doesn't respond, the escalation policy passes the alert to the backup. It can also be helpful to give everyone access to these tools before their first shift, so nobody hunts for a password at 3 AM.

How do you keep on-call sustainable?

A fair on-call load

Nights, weekends and holidays should be shared evenly across the rotation. If someone has a family event, an override lets a teammate cover those hours without changing the whole schedule. Holiday shifts work best when they're planned early, as Spike's post on on-call during holidays explains.

Alerts worth waking up for

An alert that rings a phone at night should need a human. Alerts that fix themselves, repeat for the same problem or don't affect users are the first ones to tune or turn off. Grouping related alerts also means one failure sends one alert, not ten. Spike's guide to reducing alert noise covers this step by step.

Recognition for the extra work

On-call is extra work outside normal hours, so it's fair to reward it. Companies usually pay a fixed amount per shift, pay overtime for time spent on incidents, or give time off in lieu, like a day off after a week on-call. The right choice depends on your company and local labour laws, so it's worth agreeing on it before the first shift.

5 on-call management best practices

1. Alerts work best when they carry context: the service, the latest deployment, and the runbook. That saves the person on-call the first ten minutes of digging.
2. Written expectations make on-call much easier to settle into. A short on-call policy can cover how fast to acknowledge an alert, what to keep nearby during a shift, and when to call for help.
3. New engineers usually find their feet faster with a shadow shift or two before they go on-call alone. They get the same alerts as the primary but only watch and ask questions.
4. Handovers go more smoothly on a weekday morning than on a Friday evening. Both people are awake and online, and any open issue can be passed on with time to talk it through.
5. It can be useful to review the on-call load once a month by looking at alerts per person and night-time alerts per shift. If one person carries far more than the rest, the rotation probably needs a tweak. Spike's shift reports make this check quick. A test alert outside office hours each quarter is worth adding to the same routine, since it shows whether phones ring and escalations still fire.

How does AI help with on-call?

AI takes the small, repeated questions off the person on-call and gets them up to speed faster when they're paged.

In Spike, anyone can tag @Spike in a Slack channel like #on-call and ask "who's on call for payments?". Spike replies with the primary and secondary and when their shifts end. Overrides work the same way, and a plain-English request creates the override and confirms it in the channel. Engineers in Claude Code or Cursor can ask the same questions through the Spike MCP server.

When someone is paged into an incident that's already underway, Spike AI gives them a short summary of the alerts, the timeline, and what's been tried.

What should you look for in on-call management software?

Feature	Why it matters	In Spike
Rotations and layers	Covers every hour without manual planning	Yes
Overrides	Lets people swap shifts in seconds	Yes
Phone call alerts	Useful for critical alerts	Yes
Escalation policies	Calls the backup when the person on-call doesn't respond	Yes
Slack and Teams	Lets people check and change on-call from chat	Yes
Shift reports	Shows who carried the pager and for how long	Yes
Free trial	Lets you test it on your own alerts	Yes, 14 days

Integrations matter as well. On-call software works best when it receives alerts from the monitoring tools you already use (like Datadog or Prometheus) and keeps tickets in sync with tools like Linear. Check out Spike's integrations.

Set up your on-call schedule today!

On-call management comes down to one promise: when something breaks, the right person hears about it and knows what to do. The rotation models, escalation policies, and best practices in this guide all serve that promise. None of it needs to be perfect on day one. A single rotation for the most critical service with a backup on every shift is already a working setup.

Spike lets you build that first rotation, connect an escalation policy, and test phone call alerts in one place. Try it free for 14 days.

FAQs

How long should an on-call rotation be?

A week is the usual length. Groups of fewer than five people may prefer 2-day or 4-day shifts, and anything longer than two weeks is hard to sustain.

What does an on-call schedule look like?

A common one is a weekly rotation with two people on-call each week: a primary who gets alerts first and a secondary as backup.

How many people does an on-call rotation need?

Five or more people make a weekly rotation comfortable, since each person is on-call about one week in five. Google's SRE book suggests eight for a single-site team that runs a primary and a secondary.

Should non-technical team members be on call?

Yes, for certain incidents. Support teams can handle customer updates during an outage while engineers work on the fix.

Is on-call management the same as incident management?

No, it's one part of incident management. It decides who responds to an alert, while incident management covers everything from detection to the review after the fix.

What metrics should you track for on-call?

Mean time to acknowledge (MTTA), mean time to resolve (MTTR), escalation rate, and shift load per person. Together, they show how fast alerts get answered and whether the load is fair.

Need help setting this up?

Write to support@spike.sh, or book a call with our team at spike.sh/demo, and we will help you set up Spike.

Read this online: spike.sh/oncall-guide