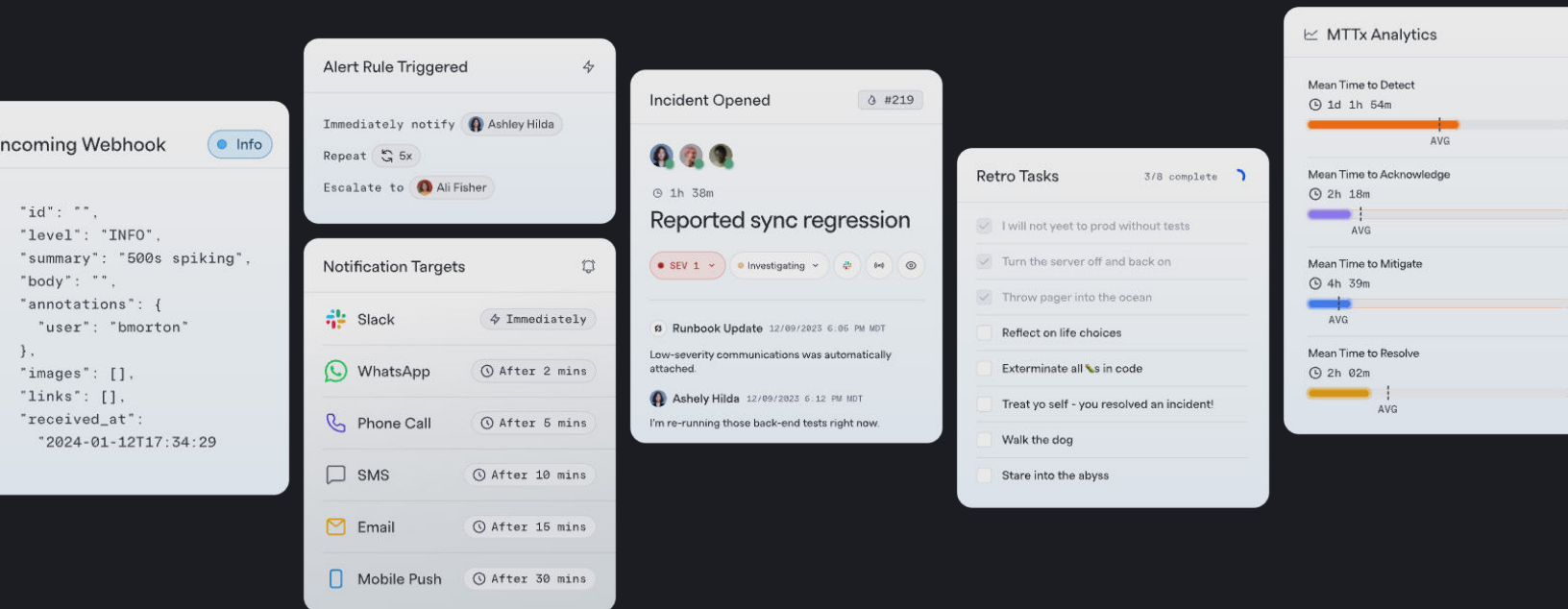




The Modern Guide to Alerting & On-Call Management

A buyer's guide to choosing an alerting solution based on insights from 500 engineering leaders



Survey conducted by



Alerting and on-call management are in dire need of innovation

Businesses around the globe are looking to streamline costs by using fewer solutions that deliver more value and efficiency. Teams now think twice about tools that used to be a no-brainer — not only because the CFO is asking them to, but because they're hopeful that a newer, more innovative tool might help them do their jobs better.

Meanwhile, in the engineering world, the major alerting players have rested on their laurels for years, failing to keep up with how software is built and run today. A more modern solution should embrace service ownership, letting teams write the rules for their alerts, on-call schedules, and incident response to deliver so much value that their budgets stretch further.

After surveying 500+ engineering leaders, we have a handle on the problems with the stagnant status quo of alerting. This buyer's guide digs into the ways current tools could (and should) be better and what engineering leaders want in an ideal solution.

After reading this guide, you'll know:

- The modern standard for alerting and on-call management tools
- The features and functionality most buyers want
- The questions you need to ask vendors as you shop for a new solution

Who we asked and what we heard

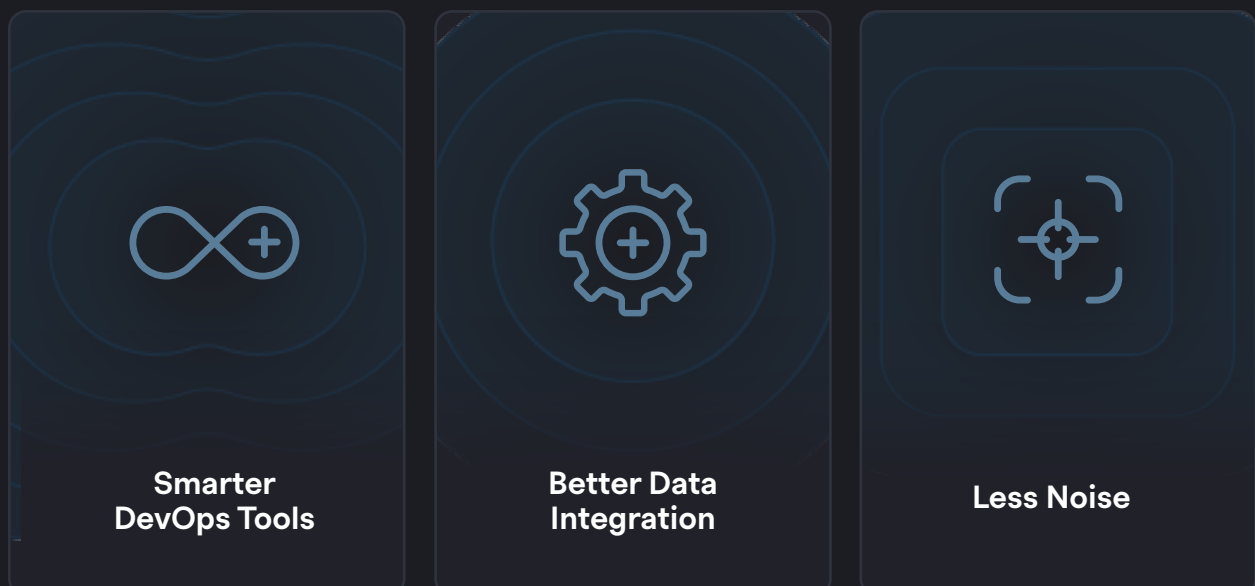
To learn what matters to DevOps and SRE leaders, we surveyed 500+ people who manage alerting and on-call for their organizations.

Respondents come from organizations large and small, with the largest group (39%) from organizations between 1,000 and 5,000 employees. More than half (53%) classify their industry as Technology, and the most common titles include:

- Director of Infrastructure (10%)
- Director of Cloud Engineering (9%)
- VP of Software Engineering (8%)
- Director of DevOps Engineering (6%)

Here's what these leaders all have in common: They deeply understand their organization's incident management processes, and they're closely involved in decision-making related to incident response. They shared with us what they want to see in a modern standard for alerting tools.

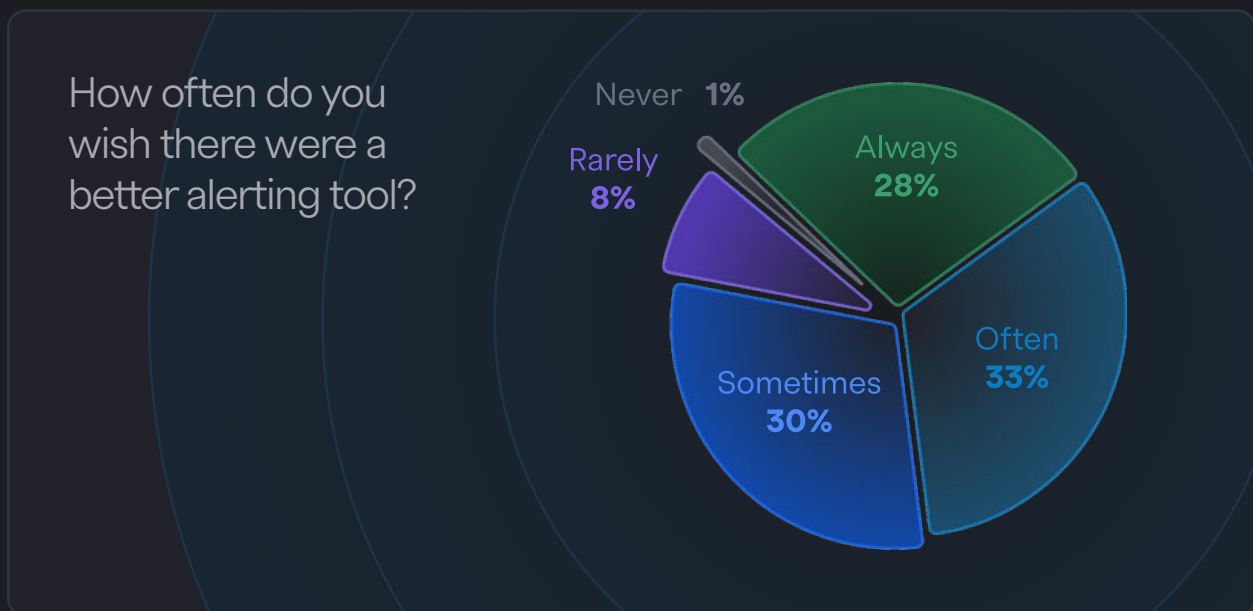
Here are the big-picture takeaways.



1. They want more out of their DevOps tools

Respondents want to see better alerting solutions on the market. In fact, 61% say they always or often wish there was a better alerting tool (and another 30% said they “sometimes” wish for the same thing).

The vast majority of our respondents (90%) said they would like to see a consolidation of tools in the observability, alerting, and incident response spaces. They’d also like an alerting tool that helps on-call engineers kick off the incident response process, with 94% interested in this kind of solution.



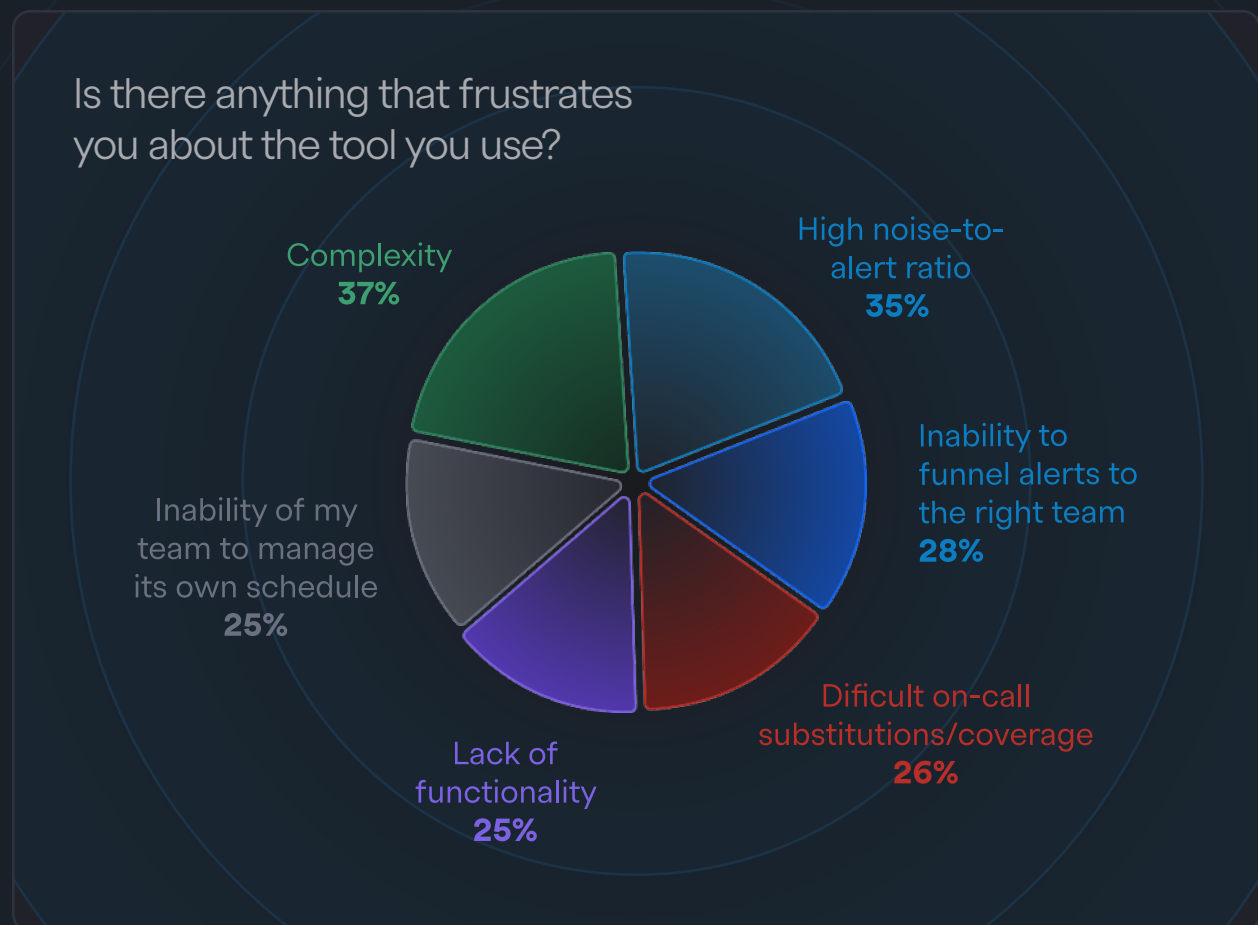
2. They want better data integration

Better alerting and incident management start with your data. More than half of engineering leaders (56%) say there are data streams they’d like to feed into their alerting tool but can’t due to provider limitations. Similarly, 54% of respondents note that the ability to ingest any stream with an API would be attractive in an alerting tool. interested in this kind of solution.

3. They have difficulty sorting out the noise in complex tools

Leaders' biggest frustrations with their current alerting tool are complexity (37%) and a high noise-to-alert ratio (35%). Another 28% say that they're unable to funnel alerts to the right team.

In short, engineers are dealing with alert fatigue due to too many false alarms, which leads to lower productivity and higher turnover. Leaders are ready for better solutions that let engineers automate more, stress less, and increase their impact.



The “old way” versus the “new way”

Today’s alerting tools haven’t kept pace with the rate of change in software. The major players were built as monolithic applications over a decade ago when we didn’t have the cloud and containerization capabilities we have available now. These early solutions were built during a different era of software, and they haven’t evolved to match how we build and manage software today.

These tools have a “systems first, people second” architecture, treating services as the primary object. Plus, in certain tools, it’s next to impossible to create and refine alerting rules, so engineers receive thousands of needless alerts on the front end and struggle to parse through incident data and make sense of it on the backend. However, teams desperately want to respond to incidents with a focus on people instead of systems. Put bluntly, legacy alerting tools simply aren’t built to make that happen.

Here are the challenges you’ll see if engineers stay stuck with this “old way” of alerting:



Disparate alerting and incident management tools



Siloed and incomplete data



Manual processes and systems



Alert fatigue

✗ Disparate alerting and incident management tools

An alerting tool that provides little to no additional value post-alert is a lot like an expensive smoke detector. Sure, it can sound an alarm to tell you there's a fire. But wouldn't you rather have a tool that will also contact the fire department?

Most alerting providers on the market don't effectively make the pass-off from "something's up" to "call the team." They alert large swaths of people and automatically create an incident or even multiple incidents (before anyone can verify that it's a real incident). This means incomplete or messy data because of the likelihood of duplicate incidents and the lack of connection from the initial event to the resolution.

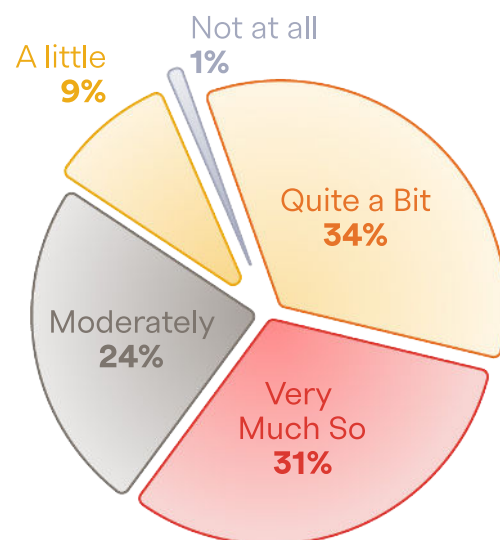
When you can track the full lifecycle of every incident from beginning to end, you actually get a better idea of a lot of things:

- Which parts of your software have the most false alarms
- Which teams or services are dealing with the most incidents
- What part of the response process is taking the longest

This intel gives actionable, clear ideas about how to improve your people, processes, and systems.

But the current point solutions just aren't cutting it. As you can see below, there's a strong desire for the consolidation of observability, alerting, and incident response tools, yet engineers are still dealing with the four following drawbacks.

Would you like to see a consolidation of tools in the observability, alerting, and incident response spaces?



✗ Siloed and incomplete data

Monitoring volume increases over time, either through previous incidents or new information. Many engineering teams create new alert rules in their monitoring system for each incident that occurs, not knowing (or easily able to see) if a rule like that already exists. As a new team or engineering leader, it's really challenging to understand what's valuable when looking at the list of thousands of rules that have been created in the past and never really cleaned up.

This tech debt adds up fast, and it's tough to pay down.

When teams struggle to refine alerts in their provider or keep clean data, they often resort to workarounds — and these configurations create a fundamental problem because it's difficult or impossible to answer questions like the ones below without the full picture of your data.

- Why did your team get 30 phone calls when there were only five real incidents?
- How many alerts for this system were just noise?
- Has [Employee Name] been on-call for each of the last few SEV1 incidents?

✗ Manual processes and systems

This status quo also drops tons of in-the-moment manual work on people, especially the on-call engineers getting alerted. They're creating tickets, rallying the team, updating status pages, and even confirming whether an incident is truly an incident as defined by your organization. (Solutions like PagerDuty, for example, automatically open an incident for every alert!). This keeps on-call engineers chained to their laptops and phones because of just how much quick, manual work they have to do when any alert comes through.

Manual analytics are time-consuming and laborious, too. Without the incident analytics you get by tracking alerts from ring to retro, leaders painstakingly track each incident to

find patterns and trends in their incident response and help engineers spend their time better. You don't get the insights you need to analyze actionable alerts versus those that aren't actual incidents.

Real-world customer stories based on early Signals customers

- “Using the API for alerting in OpsGenie is a manual process. The observability team pointed out that Atlassian has no roadmap for OpsGenie. Anytime we have a problem, trying to get through to support is my least favorite process in the world. It took me ten months to get the invoice from them when I inherited the relationship!”
- “One thing we want to track is: How did this incident start? What's the trigger for this incident? We have categories for “a customer told us” or “Shipt person saw something, said something.” Only about 10% of incidents started because we got an actual alert. Isn't the whole point that we get alerted when something is broken?”
- “What could that tell or show us if we had something already integrated with our incident management tool, like Signals? Are people just so over the umpteen thousand alerts they're getting in OpsGenie? It's too much for them to handle, so they're just ignoring them? That team is a convert now. They're clamoring to get on Signals and already love FireHydrant.”

Alert fatigue

The result of these deficiencies is a flood of calls, notifications, and texts to teams that don't need them — 71% of SREs [report](#) responding to dozens or hundreds of non-ticketed incidents per month. And, of course, you have a harder time figuring out where your team should invest their efforts and where those efforts are paying off.

This all adds up to exhaustion, cynicism, and negative self-evaluation — [the telltale signs of tech burnout](#). Engineers wind up stressed out and burnt out, and organizations feel the pain in the form of lower productivity and higher turnover, a problem 62% of IT leaders [say they've experienced](#).

It's **2.5x** more expensive to hire a new employee than to reskill a current one.

Source: Closing the Tech Talent Gap | LinkedIn Learning

The “new way” of alerting

In light of those challenges, it's time that DevOps teams had a [more modern alerting solution](#) that matches how they work. It's time for a tool that doesn't just alert but helps on-call engineers kick off the entire incident response as efficiently and consistently as possible.

Now that we understand what's not working, let's dig into what leaders really want.

“All of the things we strive to do better in incident management — respond quickly, learn from incidents, invest in reliability — only get better with native alerting.”

— **Robert Ross**, CEO & co-founder, FireHydrant

What buyers want: Features and functionality

Here's the bottom line: Engineering leaders want much, much more out of their alerting and on-call tools.

91% of engineering leaders
wish there were a better alerting tool.

What would they like to see in that better alerting solution in a blue-sky, perfect world? We asked our leaders about the features they wished they had. Here's what they said.

What buyers want:

The right data in the right place

Our engineering leaders clearly believe that better alerting starts at the source — in this case, that's the upstream data sources that feed into your alerting tool.

More than half (56%) of respondents say that there are data streams they would like to feed into their alerting tool but can't because of provider limitations. Along those same lines, the ability to ingest any data stream with an API is the second most attractive feature for an alerting tool (54%).

The “old way” of alerting means that outdated and misconfigured rules accrue, and every alert creates an incident. You’re left to figure out what’s real and actionable versus what’s not.

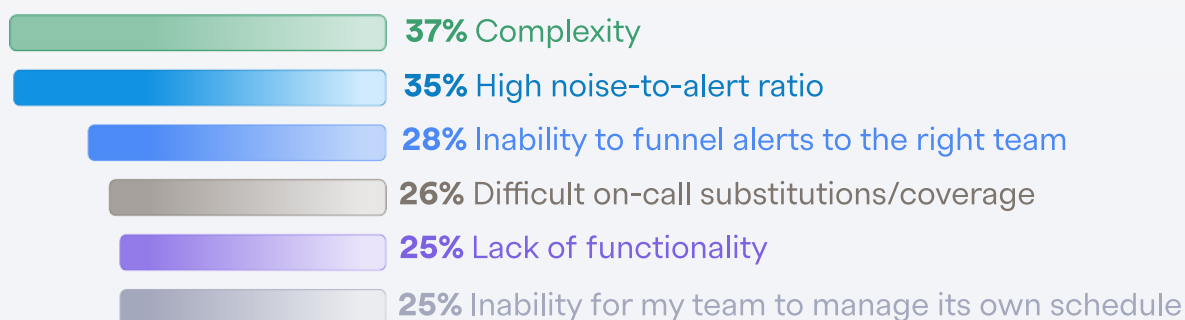
But picture a new standard for alerting — one that lets you send any data you want to your alerting tool using an API and differentiates between which alerts are important to you specifically. That way, your solution is the one drinking from the fire hose rather than you and your teammates.

What buyers want:

✓ Streamlined, fine-tuned alerting — and more value

In addition to helping maintain clean and accurate data, the ability to fine-tune alerting rules also helps address the daily challenges and organizational costs of better incident management and lower alert fatigue. So it makes sense that streamlined, fine-tuned alerting is one of the features leaders want most.

Is there anything that frustrates you about the tool you use?



When respondents were asked what frustrates them most about their current alerting tool, complexity (37%) and a high noise-to-alert ratio (35%) came out on top. These two answers go hand in hand: A solution that's too complex keeps teams from limiting their alerts based on what's actionable for them and inundates them with noise and false alarms.

Organizations waste time and work when the right data isn't getting to the right place thanks to bad or missing data and faulty routing. The top three ways legacy tooling causes additional work are as follows:

- 30% say work has been duplicated because more than one incident was opened.
- 29% say they've spent time deleting duplicate or incorrect data created by the tool.
- 27% say they've had to immediately close an incident because it wasn't truly an incident.

If that's not enough, organizations also aren't getting the bang for their alerting buck that they could be. Because fine-tuning alerts isn't easily possible, they buy extra seats that they don't need, just so certain folks can have "view-only" access — in fact, 6 out of 10 respondents say that nearly everyone in engineering has a paying seat in their alerting product.

The next evolution of alerting doesn't just create an incident for every alert. (There's enough of that with the current monolithic players.)

Next-level alerting (like [Signals](#)) makes a clearer distinction between alerts and incidents and [embraces service ownership](#). Teams write their own alerting rules, on-call schedules, and escalation policies. When an alert comes in, engineers can perform their own triage to verify incidents, graduate real alerts to incidents, and route team-specific alerts to the right teams (not services). This means all-around better incident management, straight from your alerting tool.

What buyers want:

Analytics to improve incident response

In-depth analytics — on not just one incident, but incidents over time — are key to improving the people, processes, and systems within your business to make it more reliable. Engineers crave detailed, aggregate data about how often their systems go down, where processes are breaking, and the level of accuracy and clarity their status pages present. Unsurprisingly, buyers are eager for their alerting tools that offer in-depth, comprehensive analytics.

Nearly half (48%) of respondents say they want an alerting solution that helps them better understand the overall health of their systems. They also want a tool that would help them learn:

- **Areas for improvement** in people, processes, or products (46%)
- When they need to adjust their **escalation policies** (42%)
- Where and what **technical investments are needed** (41%)

Engineering teams don't want an alert to be just an alert — as we've said, this amounts to an expensive smoke detector. They're eager to understand their organization's incident response, including metrics like alert-to-noise ratio and mean time to detect, so they can improve their processes and policies.

Unsurprisingly, when asked which features would be attractive in an alerting tool, **respondents' most popular response (55%) is analytics that allows for improvement over time.**

Leaders know the importance of improving reliability and learning from incidents. But without an alerting tool that tracks the entire lifecycle of every incident, it's hard to prove that you're making a difference — and then that makes it hard to get headcount, money, and resources for your team. Your only hope is tons of manual data manipulation, looking for patterns to find problem areas in your systems.

By tracking and reporting on data from ring to retro, leaders can spot patterns, like which teams or engineers are repeatedly responding to SEV1 incidents, for instance, to help lower burnout.

Individual contributors feel less fatigued because they can stop spinning their wheels to figure out what's real and focus on incidents that matter. Plus, they see the improvements they're making in real-time, so they know they're making an impact.

What buyers want:

Easier on-call scheduling

When we asked about frustrations with their current alerting tools, **26% cited difficult on-call substitutions and management**, and another **25% pointed to their team's inability to manage their schedule**.

These stats signal an interest in an alerting solution that lets each team [manage on-call schedules](#) and escalation policies on their own terms. In some legacy tools, on-call schedules have been extremely challenging to adjust; to simply get a shift covered, an admin has to adjust the schedule on a global scale. This makes it hard to, say, get coverage for an hour to pick up a sick kid or take an unexpected long weekend. Engineers are craving an on-call solution that keeps flexibility for the realities of life as a priority.

This approach doesn't eliminate the ability to page by service, either — with a tool like Signals, the service catalog is what ensures each page gets routed to the correct team. You can ingest multiple signals, but you still get to [write the rules](#) about which alerts call for a page, and you get greater flexibility if you want to page yourself or a specific person when a certain type of alert comes in.

Keeping configuration in the hands of an admin also makes it easier to trade on-call shifts and schedules. These quality-of-life updates to alerting make it much easier for engineers to integrate their lives with their work — and with [tenuous engineer turnover](#), scheduling is no small thing.

What buyers want:

✓ Consolidation of tools

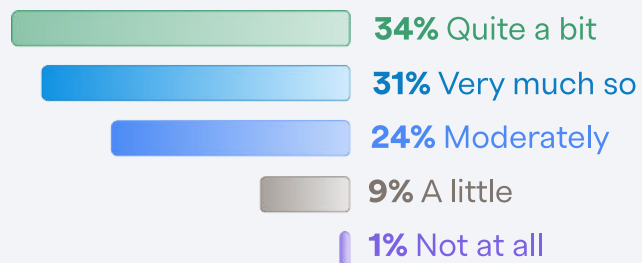
When your alerting tool is separate from your observability and incident management solutions, your team ends up doing a whole lot of “swivel-chairing.” Engineering leaders want something else — they want their tools to work together more closely.

Nearly all of our respondents (89%) say they want to see a consolidation of tools in the observability, alerting, and incident response spaces. Instead of getting an alert from one solution and then jumping to another tool entirely (and later dealing with the unwieldy data manipulation needed to scrape together any analytics), they want a single platform that covers the incident lifecycle end to end.

They also want an alerting tool that does much more than sound the alarm; they want automation and guidance on their immediate steps. It’s possible to offer them a tool that handles the:

- Automatic creation of communication channels
- Communication and sending of updates
- Tracking timeline information

Would you like to see a consolidation of tools in the observability, alerting, and incident response spaces?

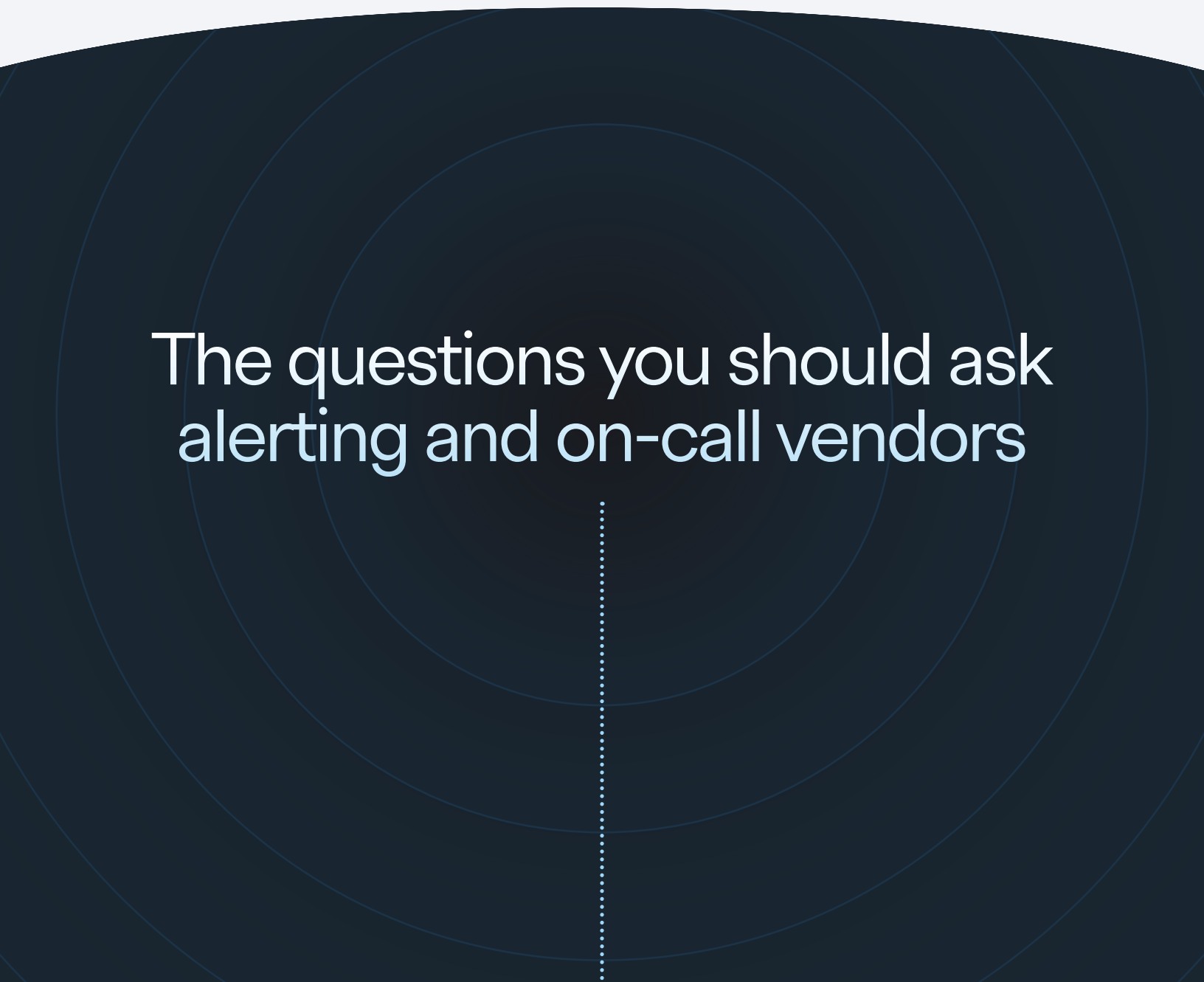


A whopping **94%** of our respondents say they want an alerting tool that automatically helps them kick off the incident management process.

Imagine an alerting tool that lets you kick off an incident from your phone. **When you get an alert sent to your phone, you can declare an incident via text message, and arrive at your computer to find your Slack channel created and your status page updated.**

FireHydrant co-founder and CEO, Robert Ross, calls a tool like this “a fast lane to receive alerts and immediately run processes.”

That's the future of alerting, one platform from ring to retro.



The questions you should ask alerting and on-call vendors



- ☐ 1. Can this tool increase our upstream data without increasing noise?
- ☐ 2. Does this solution embrace service ownership?
- ☐ 3. Does this tool limit redundancies and duplication in incident management?
- ☐ 4. How will this tool add to our understanding of our systems and processes?
- ☐ 5. Is the solution's price worth the value we get from it?

If that list of ideal alerting features resonates with you, you might be anywhere on a scale from daydreaming about a better tool to actively looking for a new solution. Either way, you should know what to look for in your next provider — and what questions to ask the alerting and incident management vendors you consider.

Our research revealed that these six questions are the most important ones to ask.

1. Can this tool increase our upstream data without increasing noise?

We've reported it already, but it bears repeating — most engineering leaders would like to feed data streams into their alerting tool, but their current provider doesn't allow it. Look for vendors that can:

- Ingest any data stream with an API
- Allow teams to only the alerts they want downstream

This will decrease noise and alert fatigue by leaps and bounds.

2. Does this solution embrace service ownership?

It's time to call it quits on some legacy providers' "systems first, people second" mentality. Engineering leaders agree — they want alerting tools to allow for rules focused around teams rather than services. When shopping for a new solution, seek out architecture that's people-first — one that actually treats humans as the primary object that's notified and lets you skip the workarounds and hacks.

3. Does this tool limit redundancies and duplication in incident management?

With too many alerts (and not enough fine-tuning), you will see duplicated efforts due to multiple incidents and false positives. Instead of automatically creating an incident based on every alert, which means you wind up with multiple "incidents" for the same event, look for vendors that give your team more control over triage and escalation policies, so you can manage incidents better and more efficiently with their solution.

4. How will this tool add to our understanding of our systems and processes?

When your alerting solution tracks each incident throughout its lifecycle, you have a more complete picture that can make it easier to find the root causes of your incidents and improve your people, processes, and systems. In short, you can make alert data actionable. While talking to vendors, picture a day in the life using their solution — will their tool make it easier to understand what percent of alerts became real incidents or how fatigued engineers are? This will give you a clear line of sight into the analytics that will improve processes and, ultimately, drive engineer retention and reliability improvements.

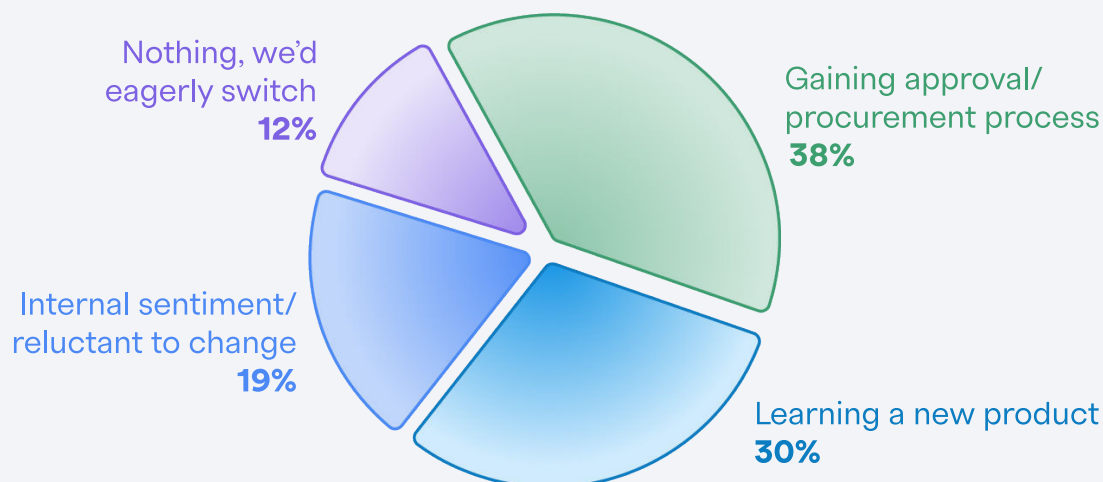
5. Is the solution's price worth the value we get from it?

Another way of asking this question is to look at your current solution and ask yourself, “Am I paying too much and getting too little?”

Organizations can only stomach the prices set by major alerting players because they think, “It’s just the cost of doing business.” But what if changing your solution could actively add value to your alerting and incident management processes — wouldn’t you want to get more for the same money? We thought so. Vet your vendors with value in mind.

What stands in the way of finding a new solution?

If there were a better alerting solution on the market that was also more cost-effective, what would stand in the way of switching?



Clearly, engineering teams and leaders know that how they alert on, communicate about, manage, and learn from their incidents could be better. They're conscious of the features that would set apart a solution that's built for this era of software.

So which obstacles stand between them and stepping into the future of alerting? We asked respondents for the top barriers to switching to a better alerting solution that would also be more cost-effective.

For most of our respondents, it's gaining approval and facing the procurement process (38%). Brian Trowbridge, FireHydrant's VP of Sales, backs this up, noting that one of the first questions prospects ask is, "What's the migration cost going to be?"

It's a valid question. Our answer? Not as much as you'd expect. For what you're paying for your alerting solution right now, you could be getting a full suite that covers the entire incident lifecycle: alerting (so what you already pay for), plus automation tooling that helps you resolve incidents faster.

“ The pain of switching away from a solution is drastically reduced when you switch away to something that comes with more value for the same cost.”

— **Robert Ross**, CEO and co-founder, FireHydrant

That's what we're doing with [Signals](#). You get more value — alerting plus incident response in one tool. Plus, with bucket-based pricing, you're paying for alerts instead of seats, which could reduce your bill by half. To get a quick estimate, [explore our cost calculator](#).

“ Just recently, one of my teammates told me they couldn't imagine not having FireHydrant. This speaks to how impactful it has been in our organization.

We are trying out Signals now and are excited about using an on-call rotation and scheduling product that's still being actively developed. Our current provider has not made significant updates in years and has settled into maintenance mode.”

— **Anthony P.**, Signals Beta Customer at Enterprise Company

Migration cost isn't just about budget, obviously. It's also about the time and productivity costs organizations pay when changing solutions. So it makes sense that respondents' second most popular obstacle to switching is learning a new product (30%).

But here's the good news: There's something freeing about ditching any accumulated, irrelevant rule debt that's built up in PagerDuty. Rules that have been created and forgotten about can go away, and you can start fresh for how your team works today.

Organizations using automation to provision PagerDuty for them can point that at FireHydrant. With some simple modifications, FireHydrant customers can switch to a new solution with a pretty low operational cost of adoption; not to mention the easier on-call management, consistently meaningful alerts, and more efficient incident management.

“Firehydrant has been a game changer for our teams for creating a culture of blameless incident response. The interface (Slack) is easy to use, easy to set up, and now with Signals, brings a revolutionary approach to on-call life.”

— **Matt C.**, SRE at Computer Software Company

The third obstacle respondents cite is internal sentiment and reluctance to change (19%). Many say, “We've had our current solution in place forever.” The sub-text, of course, is “So why change now?”

To that, we say: “Because there's more value and functionality to gain with your alerting solution.”

Your current provider alerts engineers when something's broken, but that's table stakes. The tools of the future help you kick off incidents, too — and come with so much more. Think features like team-based alerting and on-call scheduling, automated incident kickoff, and analytics that help you see everything from alert-to-noise ratios and MTTX.

In short: it's a modern alerting and incident management tool buyers seek.

We're building the future of alerting— come join us

Good news: You don't have to succumb to the current state of alerting anymore.

Organizations everywhere want to do more with less budget, fewer tools, and greater efficiency. We're convinced the current reevaluation of tools is actually a good thing. It's asking teams to take a second look at alerting: what could be better and what they want going forward.

If you're in the market for a modern alerting solution that DevOps teams love, you're looking for [Signals](#), FireHydrant's new (and very cost-effective) alerting tool.

Ready to see what a modern tool could do for your company?

Check out our pricing or get a demo to see Signals for yourself. →

