splunk> + VictorOps | catchpoint™

From Reactive to Proactive: 6 Ways to Transform Your Monitoring and Incident Response

Reduce downtime and ensure performance reliability—without intensifying on-call burnout

A CO-AUTHORED GUIDE BY
SPLUNK + VICTOROPS AND CATCHPOINT

TABLE OF CONTENTS

Introductions 4

6 Ways to Transform Your Approach to Monitoring and Incident Response 10

- Step 1: Appeal to Stakeholders with a Strong Business Case 10
- Step 2: Get Your Monitoring in Order 13
- Step 3: Think Beyond the Ticket 16
- Step 4: Throw Out the Traditional Runbook 20
- Step 5: Evaluate, but Stop Pointing Fingers 22
- Step 6: Make Reliability Everyone's Responsibility 24

Summing it All Up 26

IT'S 2 A.M. YOUR PHONE IS BUZZING. A SEEMINGLY MINOR CODE ISSUE HAS CAUSED A MAJOR OUTAGE OF A NEW APPLICATION. YOU HAVE MERE HOURS TO FIX THE PROBLEM BEFORE CUSTOMERS DISCOVER THE OUTAGE — AND HIT YOU WITH A DELUGE OF SUPPORT TICKETS.

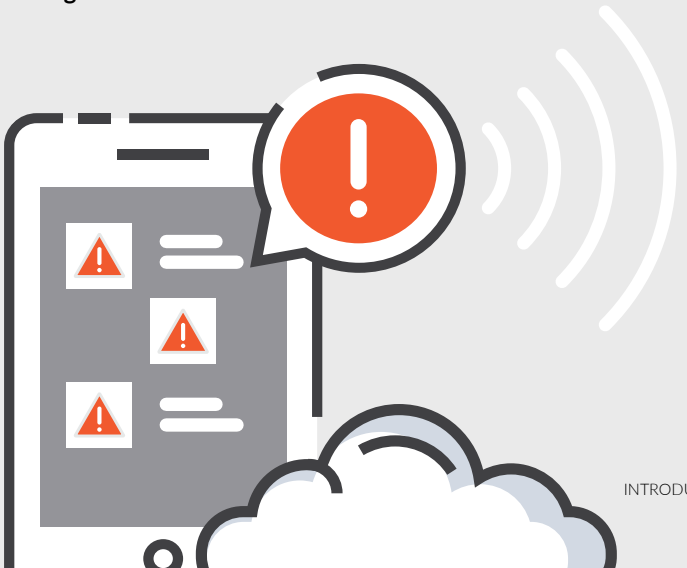
For many IT and development team members, this is an all-too-common scenario.

Why? For starters, 90 percent of development teams face pressure to release applications more quickly, according to a study conducted by Datical.[1] The demands of faster delivery in a rapid deployment environment, while a necessary evil in today's marketplace, often lead to an increased risk to availability and uptime down the line.

Compounding the issue, today's globally-focused organizations have highly sophisticated systems and infrastructures, creating multiple points of failure and making it much more difficult to pinpoint and resolve problems. A product bug can go viral in a matter of minutes. And, thanks to social media, your company's reputation can be on the chopping block just as fast.

All of this creates the perfect storm for organizations in the software and application business which now must staff 24/7 on-call teams to provide always-on solutions and services, and avoid disruptions and the backlash that follows.

This leaves many teams struggling to get ahead of incidents and manage them at scale.



The business risks of reactive incident response

Without a cohesive approach to monitoring and incident response, businesses are more vulnerable to costly downtime.

Gartner estimates[2] the business impact of downtime at $5,600 per minute, which multiplies out to over $300,000 per hour.

However, the true business impact associated with downtime is often impossible to quantify. You may be able to get a handle on how many customers you lose because your service is offline, but it's much harder to know how much business you never captured because word got around about problems with your service or solutions.

What effect(s) does downtime have on the business?

86%
of respondents
said drop in customer satisfaction

70%	Lost revenue
57%	Drop in employee productivity
49%	Lost customers
36%	Social media backlash

According to the 2019 SRE Report from Catchpoint

An outage doesn't need to be global in scale to cause serious damage to a business. Micro-outages, which are short in duration and may impact only a portion of the site or users in a specific geography, can still damage sales, frustrate customers and set engineers behind. For example, imagine an e-commerce business in the midst of Black Friday realizes that after a code push, the "Buy Now" button only appeared on some product pages, not all. While this type of incident might not topple a company, it will bruise monthly revenue numbers.

One thing is for certain: No matter how big or successful your business, you can't afford long bouts of downtime.

It's a people problem too

As business leaders work to fight downtime and increase the efficiency of their monitoring and incident response processes, what is often left out of discussions is the human impact.

Maybe some of your team members are required to fulfill on-call duties they aren't fully equipped to handle or feel frustrated by the sheer volume of work and the expectation to be available 24/7. Either way, the result is the same: burnout and turnover, both of which can be disastrous for the organization in the long run.

The 2019 SRE Report by Catchpoint explored the occurrence of incidents and the impact these incidents have on those responding to them. The majority of respondents reported some level of post-incident stress after working on an incident. The level of stress varies based on the severity of the incident and the amount of support they feel from their team and company.

Even respondents that indicate they don't experience post-incident stress reported noticing a change in mood, appetite, and desire to be social after working on an incident (all of which are signs of stress).

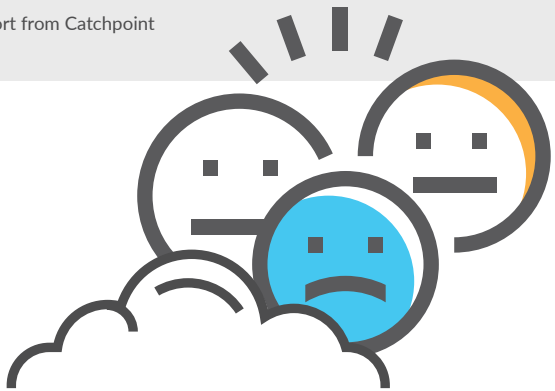
After recent incidents, do you notice a change in any of the following?

The impact of downtime on your team

52%
of respondents
said mood

48%	Concentration
38%	Ability to sleep
38%	Desire to be social
32%	Ability to enjoy things
9%	Appetite
1%	None

According to the 2019 SRE Report from Catchpoint



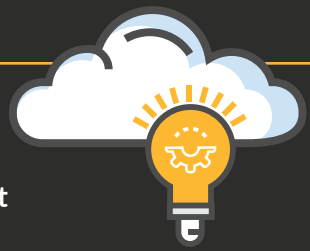
The report also found that 67% of SREs who feel stress after every incident do not believe their company cares about their well-being.

This might partially explain why the tech industry has the highest turnover rate (13.2%) across business sectors, according to a study conducted by LinkedIn.[3] The industry is already reeling from a serious talent shortage and 58% of hiring managers report struggling to find IT candidates who fit the necessary skill set.[4] IT professionals are in high demand, and companies are willing to cough up extra dough for talented engineers — but companies struggle to hold on to that talent. Many can blame a flawed incident response system that leaves teams feeling overwhelmed and burned out by on-call work. Engineers just have too many options to remain in a job that makes them miserable.

It's time to stop fighting fires by taking a more proactive approach

What's the key to successfully managing incidents, and keeping both customers and employees happy? It starts with tools and processes to streamline your operations and boost both collaboration and communication. With a stronger approach to monitoring and incident response, team members can spend less time chasing down redundant and low-priority issues and more time focused on the stuff that matters.

Even if you're not yet ready to fully commit to DevOps, adopting some of these monitoring and incident response best practices can help you modernize your incident response process.



6 WAYS TO TRANSFORM YOUR MONITORING AND INCIDENT RESPONSE:

#1 Appeal to stakeholders with a strong business case

Step one is to get buy-in that monitoring and incident response needs to change. We're talking about all stakeholders here, from the folks in the C-suite to the engineers handling incidents. Whether you're looking at buying new incident response software or simply changing processes, you need everyone to buy into the shift for your system to work.

Convincing executives you need a change

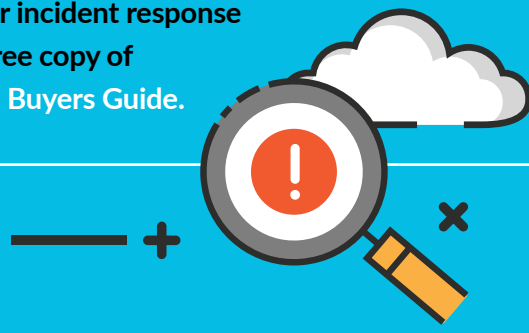
No doubt, you need executive buy-in to succeed. You may need budget approval, you'll probably need to train employees and you may need to build an entirely new team or steer manpower and resources away from other projects. Involving the C-suite early is critical, and you must be armed with data to build your case.

Come prepared to quantify the value of minimal downtime for the customer (e.g. less attrition) and the value of an improved on-call process for staff (e.g. less alert fatigue and lower employee turnover).

Here's what you should be prepared to present:

- An explanation of why you need to make a change, from a business perspective
- An outline of your plan and breakdown of any software you want to adopt
- A timeline for implementation
- Reliable financial data that shows cost savings, customer retention, and potential profits
- How the plan connects to the overall organization's goals (e.g. by reducing customer churn)
- Predictions of how the new plan or software will improve operations and the customer experience
- How you will measure success

Feel confident making your incident response software decision with a free copy of The Incident Management Buyers Guide.



Gaining buy-in from your team

While executives are mostly concerned with the bottom line, your development and IT teams will want to know how they will be affected — and whether their lives will improve as a result.

Luckily, one of the top reasons to overhaul your incident response process or to invest in new software is to do just that: make your employees' lives better!

While you definitely want to talk about big-picture benefits for the organization as a whole, if you don't tell your team members how a new software or process affects them specifically, don't count on gaining their support. The bottom line you want to impart? Strong monitoring and incident response processes + a solid tech stack makes on-call suck less.

Keys to persuasion

Whether you're talking to the CEO or an engineer, make sure you:

- Clearly state what does and doesn't work about your current protocols
- Focus on how a new tool or plan will resolve hot-button issues (e.g. you just lost a huge client because of downtime)
- Focus on the benefits of your solution — scare tactics won't work!
- Back up everything you say with data and real-world examples
- Help them understand that more efficient incident response translates into more time to innovate and deliver new services that differentiate your business

#2 Get Your Monitoring in Order

While some teams lack insight into the health of their systems, others are drowning in a sea of notifications, making them more likely to miss the root cause of these problems. The result of either scenario can be that incidents swell into serious emergencies and customers experience downtime.

To combat false positives/negatives and prevent chaos around incident response, it's important to focus on establishing the right mix of monitoring processes.

Your monitoring and alerting strategy must include:

- Setting appropriate alerts and thresholds
- Getting insight into all layers of the delivery chain
- Configuring tests to run at the right frequency
- Sending notifications to the right people
- Classifying alerts based on priority
- Establishing an escalation policy to handle high priority alerts

Steps to reduce false positives and false negatives:

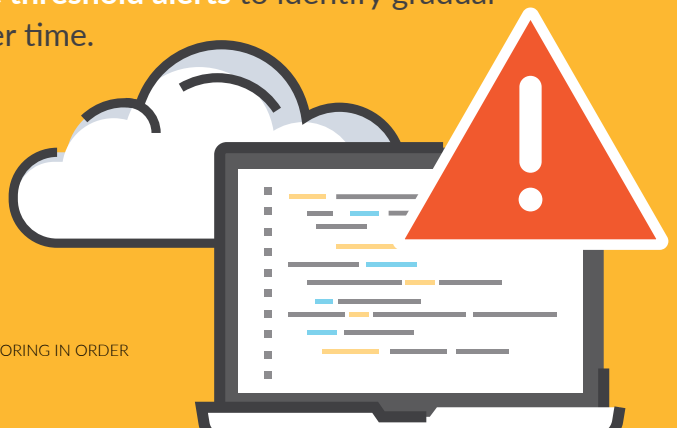
Monitor from the end user's perspective. This means monitoring from multiple ISPs on backbone, broadband, last mile and wireless networks so that you can triangulate across multiple vantage points to identify the root cause. Monitoring from a single ISP or cloud provider creates visibility gaps in your monitoring and can result in false positives. An alert shouldn't be triggered when a provider has an issue. It should be triggered when your application or service has an issue.

When a test fails, automatically run another test to verify immediately that an error condition exists and the failure was not an anomaly.

Do not trigger an alert when a failure is due to a fault within the testing environment.

Use trend shift alerts for metrics prone to constant variations.

Use specific threshold alerts to identify gradual changes over time.



Types of Monitoring:

Any or all of the following monitoring tools can be used to identify points of failure:

- **Digital Experience Monitoring** provides a wide-angle view of all the components that make up your digital services, and captures data from the end user's perspective. Metrics are collected from real users and synthetic agents to identify anomalies and incidents throughout multiple layers of the delivery chain that are impacting end users.
- **Proactive Synthetic Monitoring** simulates requests to applications and services to verify performance, availability, and reachability. It includes issuing requests to DNS, FTP and APIs, or simulating users that are accessing an application.
- **Real User Monitoring** measures performance from actual users visiting a web site. Data is collected via a script in real time.
- **Application Performance Monitoring** includes transaction profiling, tracing, and code-level diagnostics to understand how a request is mapped across servers and microservices. Metrics are typically collected from agents installed on the servers.
- **Network Monitoring** through a system that constantly monitors a computer network and sends alerts when components slow down or fail.

Ready to upgrade your approach to monitoring? Book a Catchpoint trial account or schedule a private demo: https://www.catchpoint.com/trial

#3 Think beyond the ticket

By this point you might be thinking, “My organization has an ITSM tool like ServiceNow…problem solved, right?”

Not quite.

Ticketing systems, while serving an important role, are simply not an efficient way to manage critical outages.

Think about it — the typical ticketing system requires a person to file a ticket before anyone jumps into the firefight. In the days of the employee-facing system of record, this process worked just fine. But as IT moves from employee-facing systems of record to customer-facing systems of engagement, IT problems are surfaced differently.

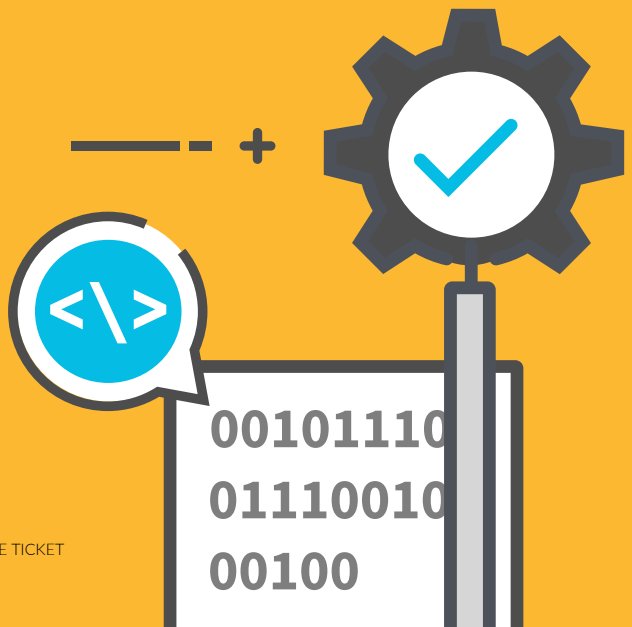
Systems of record (such as email or ERP) require employees to put in trouble tickets when things break. With systems of engagement, your customers won’t bother filling out trouble tickets — they’ll just abandon your website, go to your competitor and complain on social media. Effective response to incidents for your systems of engagement requires monitoring tools that can spot problems, paired with collaborative incident response tools that engage the right people. IT Ops, SREs and developers should not have to wait to have a trouble ticket in-hand to investigate and resolve problems.

Furthermore, from a problem resolution standpoint, ticketing systems lack automated routing and escalation policies, which are essential to reaching the right people to resolve issues quickly. Even when they do reach the “right” person, they don’t provide context and other critical information, so they tend to delay MTTR (Mean Time to Resolve). They also aren’t suited for more complex incidents. Team members typically pull up a ticket, choose a canned response and move on, but that doesn’t always work if additional information or input is needed to resolve the more complex problems.

And when it comes to the human impact, ticketing systems often lead to alert fatigue. Employees are inundated with repetitive questions and must manually respond to each one. This can become overwhelming, causing a delay in responding to more serious incidents — or missing them altogether as there is too much noise in the system to manage.

None of this is to say you should throw out your ticketing system altogether. Instead, think of your ticketing system as the hospital — if an emergency occurs, you can’t sit in traffic, waiting to get to the hospital before seeking care. Your incident response system, on the other hand, is like the ambulance. As soon as an incident is triggered, the tool can start delivering relevant, targeted alerts to the right person, with runbooks and other important documentation attached, so you can start triaging the issue immediately. Both the hospital and the ambulance serve critical — but different — roles.

INCORPORATING A TOOL THAT'S PURPOSE-BUILT FOR INCIDENT RESPONSE — RATHER THAN RELYING SOLELY ON A TICKETING SYSTEM — ENSURES PROBLEMS ARE RESOLVED QUICKLY AND EFFICIENTLY.



With an incident response tool in your back pocket, you can expect:

- **On-call management options**, such as scheduling, notifications (e.g., phone, email, SMS and push), extendable API and webhooks
- **Full mobility**, giving you complete capability (e.g., annotations, bi-directional chat, incident transparency and post-incident documentation) to investigate and resolve issues, even when you're on the go
- **A rules engine** to support efficient routing to automate and streamline alerting
- **Robust incident metadata** (e.g., runbooks, graphs and notes) for contextually-relevant alerts
- **Reporting capabilities** which are relevant to the first few stages of the incident lifecycle, including MTTA (Mean Time to Acknowledge), MTTR and incident frequency reporting that is both customizable and actionable
- **Post-incident documentation** that serves as a black-box recorder before, during and after an outage, so you gain critical insights to prevent future issues
- **Integrations** with hundreds of other tools

To learn how an incident response tool could transform your organization, schedule a personalized demo with one of our experts.

#4 Throw out the traditional runbook

When establishing a streamlined monitoring and incident response protocol, you cannot afford to reinvent the wheel with each incident. This wastes time and causes on-call team members undue stress.

This is where runbooks come into play.

A runbook provides step-by-step, clear directions for diagnosing an issue, responding to an escalating alert and resolving the problem. It requires an ongoing process of idea generation, prototyping, automation, presentation, information capture, analysis and learning.

Incidents are inevitable, and outages happen. Providing your team with contextual documents they can turn to when the heat is on helps to ensure they have the information they need to quickly resolve incidents and significantly reduce downtime.

That said, as important as runbooks are, if they're extremely time-consuming to create and maintain, they may not be worth theeffort. They become especially problematic if you must pull staff away from higher priorities that have a greater influence on the bottom line in order to follow them.

The solution isn't to throw out the runbook but to rethink it: Minimum Viable Runbooks (MVR) provide all the information you need—and prepare your team for the next incident—without taking too much time to create and manage.

How to create a minimum viable runbook

- **Build a record of effective actions.** You will need the capability to capture the conversations, collaboration and actions that led to resolutions. An incident response tool like VictorOps allows you to send alerts, actions and chats into a single incident timeline.
- **Establish a protocol**. Teams should be collaborating—whether through Slack or Microsoft Teams (all of which can easily be integrated with an incident response tool—and ensuring that all activities undertaken are sent to the incident timeline. Without that information, you won't be able to tell what is and isn't working, so set rules to ensure people are using the appropriate channels.
- **Develop a "rules engine."** Use an "if this, then that" approach to processing your incoming alerts.
- **Build action plans**. Provide clear steps to resolve issues in the future, being sure to answer: What, How, Who and Where. Ideally, this happens in a blameless post-incident review. The "What" should tell employees which tools, status sites and charts they should review before they start problem-solving. The "How" should provide the first action a person should take to fix the problem. The "Who" should detail which other team or team members to contact if they need to escalate the incident. The "Where" provides the tools and digital locations of where to record notes, update statuses, post questions, record activities and so on.
- **Make runbooks accessible.** Append incoming alerts with the MVR, so when the same incident or alert occurs again, team members have the action plan in front of them. Provide clickable links to a reference document or a detailed runbook within the alert. The goal is to provide technicians everything they need in one place, minimizing time spent searching for solutions.
- **Keep runbooks current.** Most importantly, it's key to continually analyze and tweak your runbooks to ensure the instructions you've outlined continue to work.

#5 Evaluate, but stop pointing fingers

Every incident, no matter how small, offers a powerful lesson to your team on how to work more efficiently to resolve future incidents faster. The time to learn that lesson, however, is not when the team is in the trenches working to resolve an issue. That's why it's critical to capture the actions the team took to resolve the incident, and review them in a post-mortem — or, as we like to call them, post-incident reviews.

Why the name change?

Too many leaders — and team members — view the post-mortem as a time to skirt blame or even point fingers at others. In this environment, a post-mortem is a waste of time, and rarely leads to real improvements. Chances are, you've felt personally victimized, or at least frustrated by a post-mortem, so we'll move forward with the less stigmatized term: post-incident review.

Post-incident reviews are an opportunity to learn from failures, to get to the root of issues and to continually improve how you respond to incidents. This requires a shift in thinking from "What did we do wrong?" to "What did we learn?"

This doesn't have to be a formal, complex process. It can start out as a simple review of a centralized incident timeline among stakeholders.

How to conduct a blameless post-incident review:

Gain a clear understanding of the extent of the issue. Ask:

- Which services or products were involved?
- When did we first know about the issue?
- How did we find out about it (e.g., an alert or customer complaint)?
- How many customers were affected?
- How widespread was the issue?
- How long was service affected?

Conduct an analysis of the problem. Ask:

- Where do we think the failure originated?
- What do we think caused the issue?
- Has this happened before?
- Could we have done anything to prevent it this time?

Understand the actions you took to resolve the issue. Ask:

- How did we diagnose the problem?
- What specific steps did we take to fix the problem?
- What actions did not work as we attempted to resolve the issue? Which caused more problems?
- What specifically did we do to fix the problem?

Close your discussion by looking ahead. Ask:

- How can we prevent this problem from happening in the future?
- What's one action we can do today to improve our incident response?
- Build those remediation steps into runbooks.

It's not enough to just conduct the post-incident review. Use what you learn to make improvements in your process. If you conduct these sessions just for the sake of it, and no action comes from them, your team won't take the process seriously, nor will team members learn from past failures.

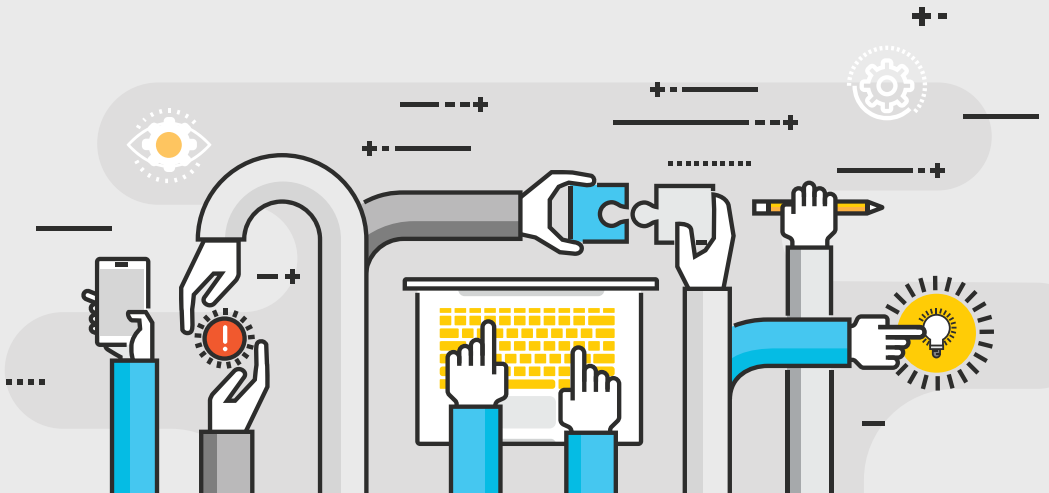
Have you ever been part of a completely useless post-mortem? If you're nodding your head, then check out this guide on why so many post-mortems fail, and how to fix them.

#6 Make reliability everyone's responsibility

According to the report by Catchpoint, the majority of SREs work in organizations with fewer than 10 SREs, and 6% are the sole SRE in their organization. No matter the SRE team size, the responsibility for protecting the customer experience and ensuring the product functions and performs as expected shouldn't rest only with that team. Every engineer plays a role in preventing reliability issues.

To ensure this, you have to create a culture of reliability — one where you hold employees accountable for proactively addressing issues and taking actions that align with the objectives of the business and the needs of the customer. Here's how:

- **Consider how you prioritize reliability.** Do you say it's important, but allocate your resources and manpower to other priorities? Are you investing in tools to ensure the reliability of your products? Or, are you allowing incidents and downtime to plague the customer experience? If reliability is actually a top priority, you need to do more than just talk about it. Given the number of changes you have in your applications and infrastructure, you are, in essence, testing in production. You need the right processes and tools in place to ensure services in production remain reliable.
- **Examine your tools.** If your team is wasting time on misleading and repetitive alerts, they aren't focusing their time and energy delivering the best possible experience to customers. You need automation tools that reduce and prioritize alerts.
- **Define reliability for your organization.** How does the ideal product perform? What would make it reliable in the eyes of the customer? Establish concrete criteria and set goals for improving the performance, functionality and dependability of a product. Examine best practices in observability to better evaluate the health of your platform and services based on externally measurable outcomes.
- **Set expectations.** Tell stakeholders the goals you have and hold them accountable for proactively helping the team and organization meet those goals. Check in regularly to monitor their progress, and make sure you acknowledge them when they contribute toward those goals. Above all else, empower your engineers to own the reliability for each of their domains or products.
- **Value feedback — even the negative kind.** Engineers must believe their efforts will be valued by the organization, so ensure you are capturing their ideas and implementing them when possible. Most importantly, encourage them to be candid with their feedback and open about their ideas for improvement. If they aren't speaking up, ask plenty of questions and actively listen to the answers, then act on what you hear.



SUMMING IT ALL UP

You're providing 24/7/365 services and solutions, and if you want to remain in business, it's critical to nail your monitoring and incident response to limit downtime and disruptions for customers. The catch is that you can't expect your employees to work around the clock and endure ineffective tools and faulty processes in order to achieve that.

Follow the lead of some of the most successful IT and development teams in the industry and start taking steps today to initiate changes that will not only help you provide better service to your customers, but will help your IT and development teams live up to their full potential.

About VictorOps

With VictorOps, "on-call" isn't a four-letter word. Our incident response software aligns log management, monitoring, chat tools and more, for a single-pane-of-glass into system health. Using this IT and DevOps system data, we support automated alerting, centralized information, and essential documentation. Teams receive context-rich notifications and collaborate cross-functionally to empower fast, efficient incident resolution—and reduced downtime. To experience VictorOps in action, check out our 14-day free trial: https://victorops.com/start-free-trial

[1] https://www.datical.com/whitepapers/survey-the-state-of-database-deployments-in-application-delivery-2

[2] Cappuccio, D. J. (n.d.). Ensure Cost Balances Out With Risk in High- Availability Data Centers(Rep.). Gartner.

[3] https://business.linkedin.com/talent-solutions/blog/trends-and-research/2018/the-3-industries-with-the-highest-turnover-rates

[4] https://business.linkedin.com/content/dam/me/learning/en-us/pdfs/lil-guide-attract-retain-top-tech-talent.pdf

About Catchpoint

Catchpoint's Digital Experience Monitoring Platform helps industry leaders ensure that every experience they deliver across the globe is monitored and optimized. The Catchpoint DEM Platform combines synthetic and real user monitoring to provide complete visibility into every layer of the modern application delivery chain, including networks, application code, CDN, DNS, services, APIs, and more. Catchpoint also offers the world's largest, most geographically-dispersed monitoring network, utilizing over 700 backbone, broadband, cloud and wireless vantage points. Customer Experience Leaders like Google, L'Oréal, Verizon, Oracle, LinkedIn, Honeywell and Priceline trust Catchpoint to provide real-time intelligence into applications and services to detect, repair, and optimize digital applications faster. See how Catchpoint can reduce your Mean Time to Detect at www.catchpoint.com/freetrial.

Catchpoint + VictorOps: Better Together

Integrating Catchpoint and VictorOps allows you to detect anomalies, predict outages and gain a deeper understanding of your system as a whole. Get the jump on problems, diagnose the issue, get the information you need in real time, and collaborate with other team members to remediate the incident. Learn more about how to up-level your approach to monitoring and incident response with our best-in-class integration: https://help.victorops.com/knowledge-base/victorops-catchpoint-integration/


splunk>
+
VictorOps
catchpoint™