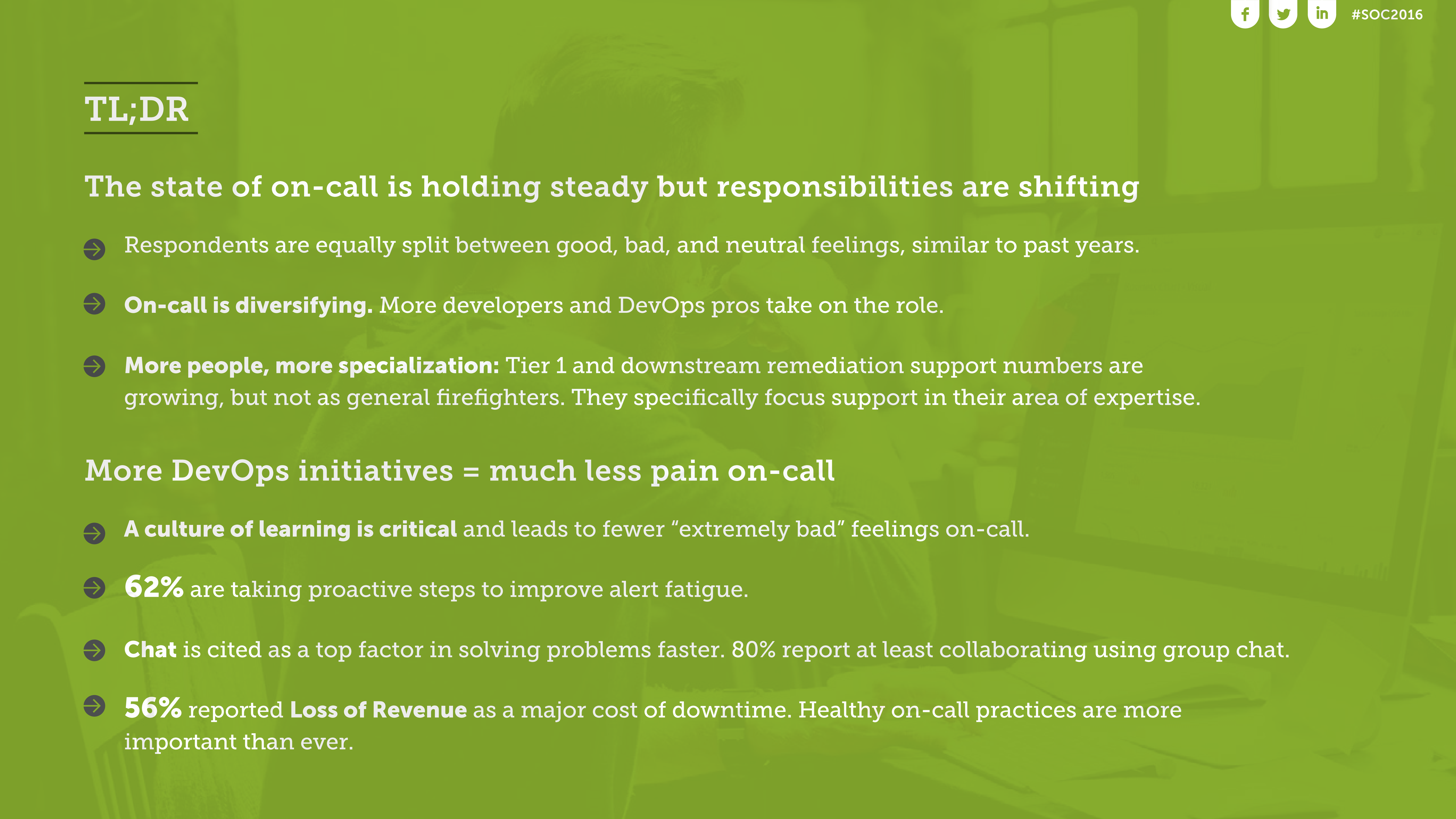
2016 VictorOps 2017

STATE OF ON-CALL

The third annual survey of the people who manage the systems and solve the problems.



TL;DR

The state of on-call is holding steady but responsibilities are shifting

- Respondents are equally split between good, bad, and neutral feelings, similar to past years.
- **On-call is diversifying.** More developers and DevOps pros take on the role.
- **More people, more specialization:** Tier 1 and downstream remediation support numbers are growing, but not as general firefighters. They specifically focus support in their area of expertise.

More DevOps initiatives = much less pain on-call

- **A culture of learning is critical** and leads to fewer “extremely bad” feelings on-call.
- **62%** are taking proactive steps to improve alert fatigue.
- **Chat** is cited as a top factor in solving problems faster. 80% report at least collaborating using group chat.
- **56%** reported **Loss of Revenue** as a major cost of downtime. Healthy on-call practices are more important than ever.

A Call to Arms: Make On-Call a Better Experience

This on-call survey was our most comprehensive to date. We wanted a deeper understanding of the ways that pressure for continuous deployment and more/better/faster resolution have impacted the state of on-call, so we delved into infrastructure, culture, costs of downtime, incident management maturity, and DevOps practices to get at the real picture of what's going on today.

The findings are not surprising. The on-call role continues to evolve to include more people and more developers and DevOps pros. Many orgs are putting key health indicators and real-time collaboration processes into place. ChatOps continues to be one of the most valuable tools for solving problems fast. And the most cutting-edge developers own their code, taking responsibility for supporting it in production.

But the real takeaway relates to the predictability of on-call sentiment. Hundreds of responders shared specific, consistent reasons for happiness or misery on-call. Extremely unhappy on-call respondents suffer from powerlessness to solve problems, excessive alert noise, horrific work/life balance, and the pain of never being able to fully relax.

Common indicators for extremely good on-call experiences include:

- **The Right Tools and Systems:** These factors include reliable teammates, good setup for alert routing and escalation, and the ability to reliably get answers.
- **Component Ownership:** Pride in their component, taking responsibility, and knowing how to fix their work were characteristics of the most advanced orgs.
- **Sense of Satisfaction:** Many respondents are wired to enjoy challenges and gain satisfaction from solving problems.
- **Culture of Learning:** These responders' orgs continuously evaluate and incrementally improve their on-call processes, while maintaining open communication channels on what is effective.

On-call can be a painful, thankless job. But it can be highly functional and satisfying. We urge you to review the report. Then if you have the power, use this knowledge to continue to improve the on-call experience at your organization. If you can't affect change, then know that there are places where the role is empowering and fulfilling.

We received over 800 responses, up from the 600 we received last year, making this what we believe to be the largest sampling of on-call practitioners in the tech space.

Who Took This Survey?

89% of respondents spend time on-call

The other 11% manage people on-call

- Respondents represent **29** different industries.

- **41% of respondents support both internal and external customers, 35%** support internal company systems only, and **24%** support external customers only.

- **Company size was equally diverse,** with **56%** from companies with <500 employees, and **44%** with 500 to 10K+ employees.

SAAS BUSINESS	13%
IT SERVICES	10%
CLOUD SERVICES	8%
FINANCIAL	8%
SOFTWARE DEVELOPMENT	8%
E-COMMERCE	7%
DIGITAL MEDIA	4%
HEALTHCARE	4%
MEDIA	4%
ENTERTAINMENT	3%
OTHER	3%
RETAIL	3%
TELECOMMUNICATIONS	3%
COMMUNICATIONS	2%
COMPUTER/NETWORK SECURITY	2%
CONSULTING	2%
DATA	2%
EDUCATION	2%
GAMING	2%
GOVERNMENT	2%
INSURANCE	2%
TRANSPORTATION	2%
AGRICULTURE	1%
ENERGY	1%
HOSPITALITY	1%
NOT FOR PROFIT	1%
SOCIAL MEDIA	1%
SUPPLY CHAIN MANAGEMENT	1%
UTILITIES	1%

The State of On-Call is Steady, With 3 Equal Subsets

But responsibilities are changing fast. More advanced organizations are shifting toward end-to-end code ownership.

Respondents fall into three groups:

32% feel extremely or somewhat bad about being on-call

30% are neutral

38% feel extremely or somewhat good about being on-call

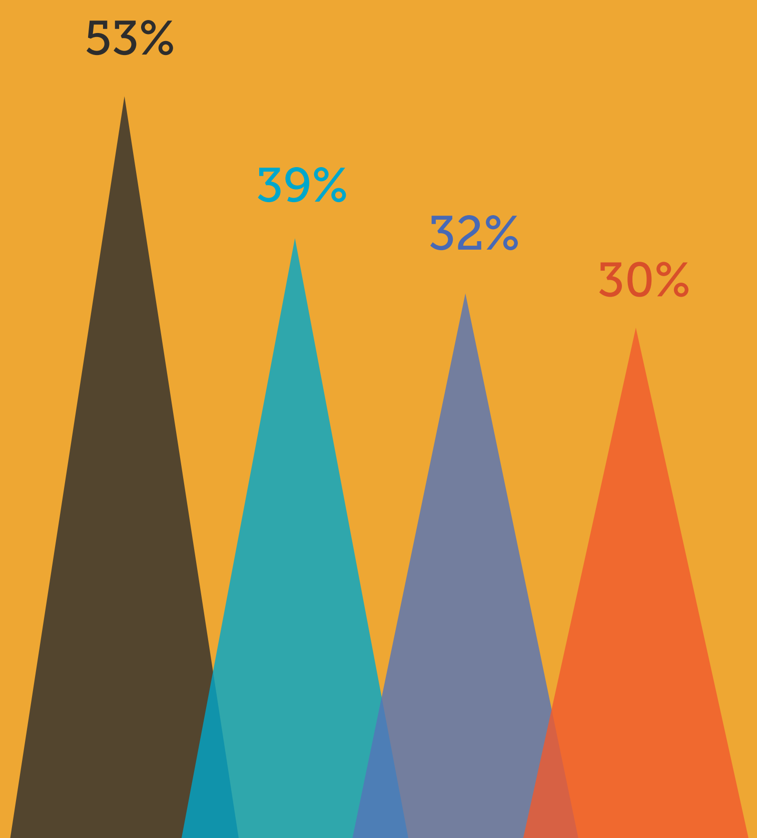
These People Hate Life On-Call

Extremely miserable on-call professionals:

- ✕ Have terrible work/life balance
- ✕ Are supporting poorly-designed systems
- ✕ Feel powerless to solve problems
- ✕ Generally hate the role

Top-reported problems:

- ▶ Alert Noise
- ▶ Lack of Remediation Information
- ▶ Inefficient Communication
- ▶ Inaccurate or Difficult Reporting



In their words...

“It makes me really nervous to handle that kind of responsibility in production, and I have to be an expert on the entire codebase, or else rely on the helpfulness of those who truly are the experts.”

“I get paged for issues that I can’t resolve; most of my time is either researching a problem that is transient or non-reproducible, or contacting a vendor. Also, I am a single mom with a toddler and an infant. It is a terrible burden to deal with incidents during off hours, and my phone often wakes them.”

“[We have] snowflake infrastructure.”

“Our monitoring tooling is crap. Other teams dump problems on us without performing any triage/investigation.”

Extremely Happy On-Call Professionals Have:

The right tools and systems:

They have reliable teammates, a good setup for alert routing and escalation, and the ability to reliably get answers.

Component ownership

They own their code and components and take pride in fixing problems that may arise.

A sense of satisfaction

They are wired to enjoy challenges and gain satisfaction from solving problems.

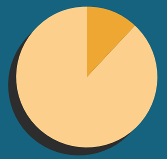
A culture of learning

They continuously evaluate and incrementally improve their on-call processes, while maintaining open communication channels.

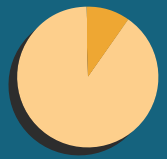
Fortunately, These DevOps Practices are Reaching New Audiences…

…while seasoned practitioners continue to hone their skills.

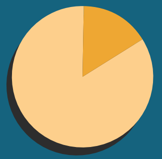
Time Practicing DevOps:



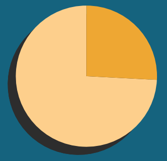
DevOps What? **(12%)**



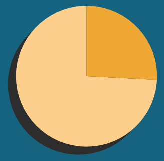
0-6 Months **(10%)**



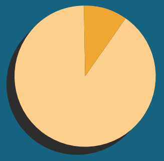
6 Months-1 Year **(16%)**



1-2 Years **(26%)**



2-5 Years **(26%)**



5 Years + **(10%)**

Same as last year, the top two most common DevOps practices are:

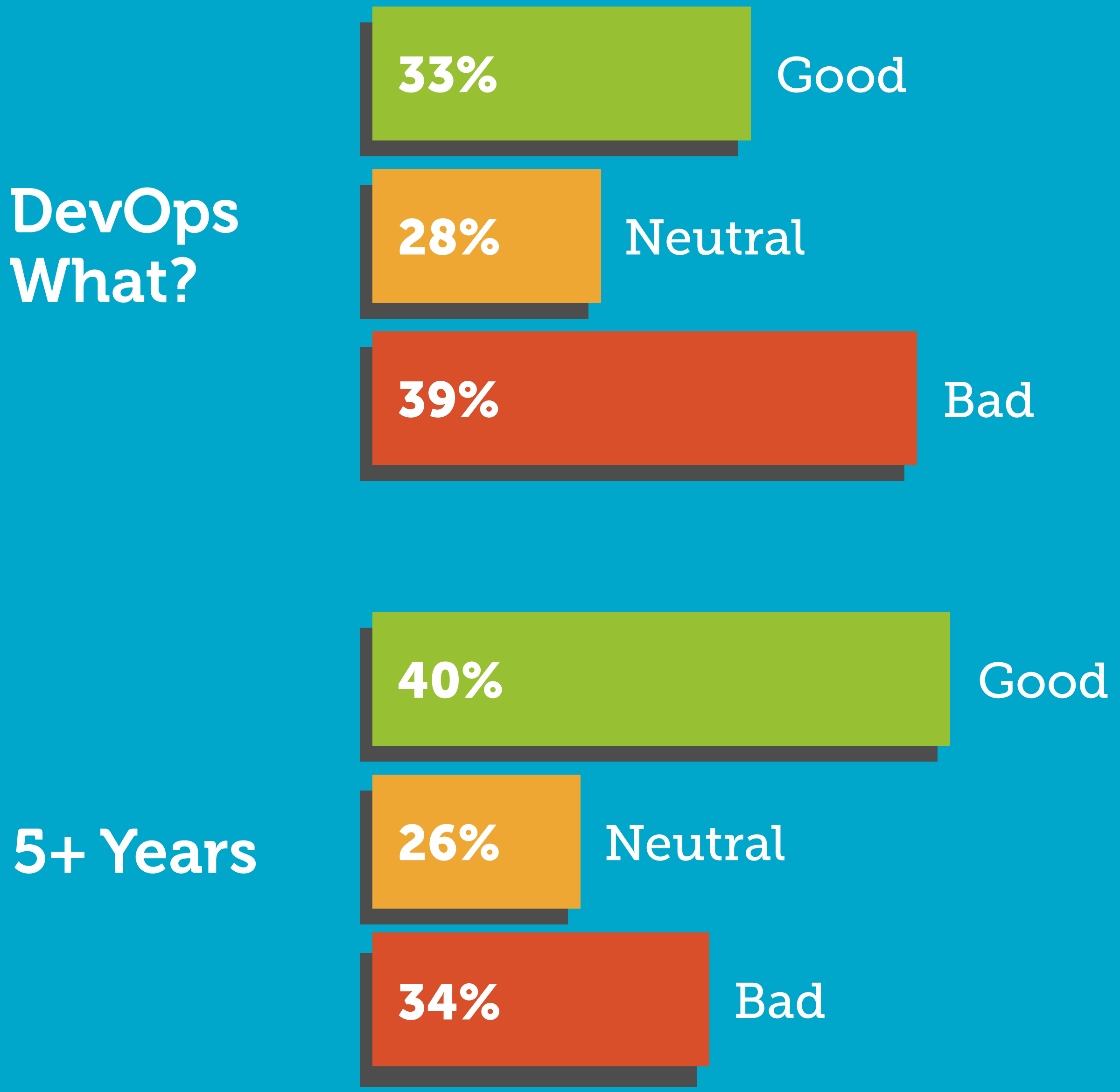
1. Automation
2. Building a culture of learning from mistakes

What has been most valuable in helping you solve problems faster?

"Broader implementation of ChatOps/DevOps practices. The consistency and standardization from Infrastructure as Code and Process Automation is significant. Over time we gain greater resource capacity and can refocus towards tasks that create more value."

More DevOps = a Much Better On-Call Experience

Those with no DevOps experience feel much worse about being on-call than those with 5+ years of DevOps experience.



In their words...

"We're moving toward independent teams with full component ownership from development to on-call."

"I like being on-call and supporting our company."

"We have the right tools and systems in place and a well-trained staff, which greatly reduces the stress level of being on-call."

"I feel responsible for the quality of the product I build. I want it to work well and create the impact it was designed for. It makes sense to have the people building the software also operate it."

More Devs and DevOps Pros are On-Call

Development and DevOps pros were more commonly on-call this year than strictly IT teams, a change from last year.

The most frequent functional areas on-call (in order) are:

1. Operations
2. Development
3. DevOps
4. IT
5. Support

On-call team size is growing:

The more people on-call in their area of expertise, the better the quality of life.

→ On average, **40 people** are on Tier 1 support, with a range from 0 to 1000 people.

→ On average, **70 people** are part of the downstream remediation process.

In their words...

> "I'm strictly the escalation point so I'm on-call all the time, but it rarely bubbles up to me. When an issue does get big enough to reach me, I need to know about the problem anyway."

Learn more about Putting Devs On-Call

Some Aggregate Alerts to Reduce the Noise

Over 60 different monitoring tools played a part alerting you. The most frequent ones mentioned:

APM	Network	Log	Web	Full Stack	Infrastructure
New Relic	Solarwinds	Logstash	Amazon Cloudwatch	Datadog	Nagios
AppDynamics	PRTG	Splunk	Pingdom	Sensu	Zabbix
Monitis		Sumo Logic	Catchpoint	Zabbix	

Incident Management tools mentioned include: xMatters, VictorOps, ServiceNow, PagerDuty and OpsGenie. **13%** have a NOC to manage incidents.

In their words…

"We use Nagios, Cloudwatch, Papertrail, and Pingdom to funnel alerts into VictorOps, which then routes to the on-call engineer."

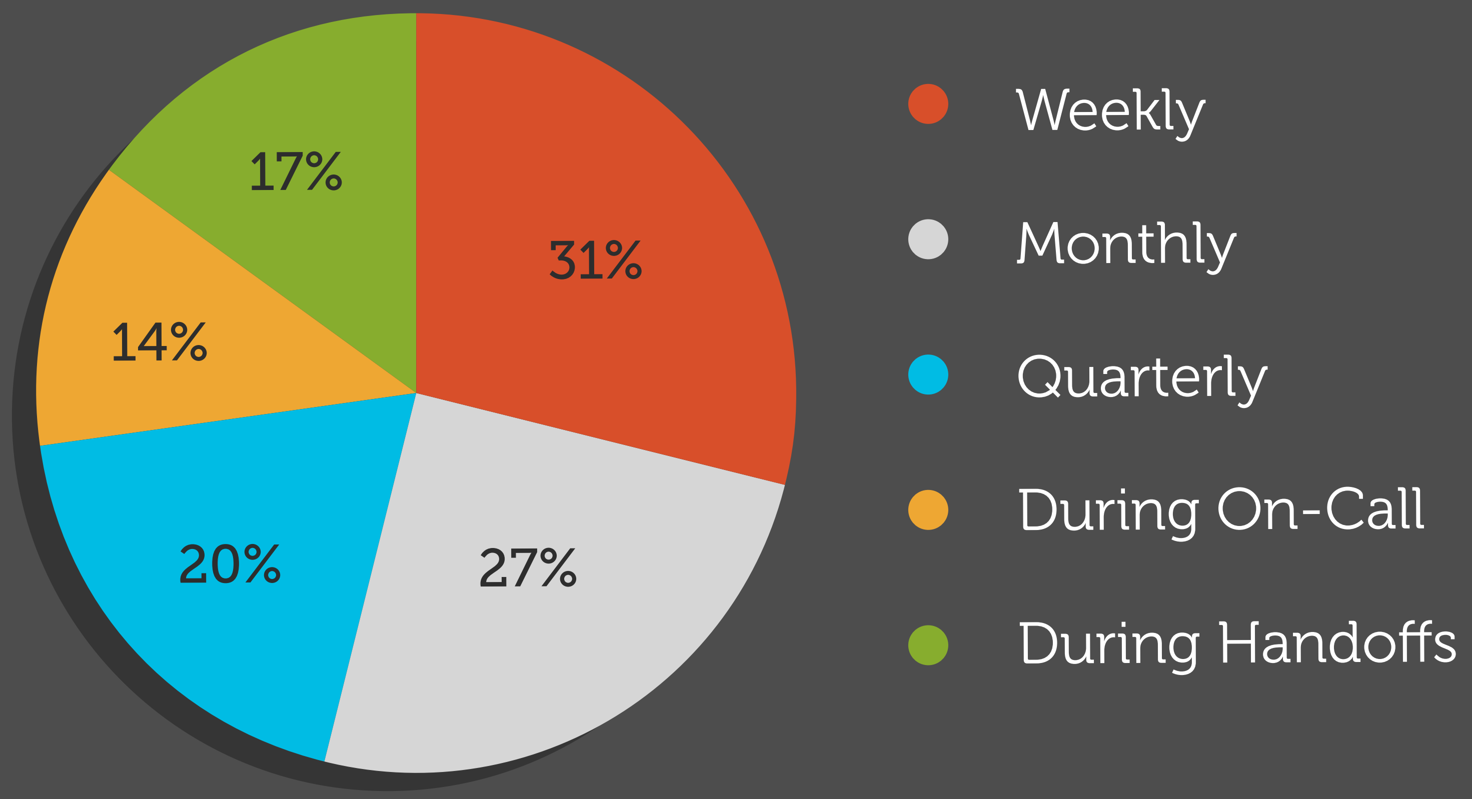
"The VictorOps app makes it easy and useful to @reply to someone as well as the entire group if necessary. Also the timeline in VictorOps…that is the first place we look to see what is going on."

Looking for tools? Get the Incident Management Buyer's Guide

62% are Proactively Reducing Alert Fatigue

- **61%** state that alert fatigue is an issue at their organization, on par with the past 2 years.
- Top remediation steps include making alerts contextual and actionable, and reducing redundancy.

How often do you take steps to reduce alert fatigue?

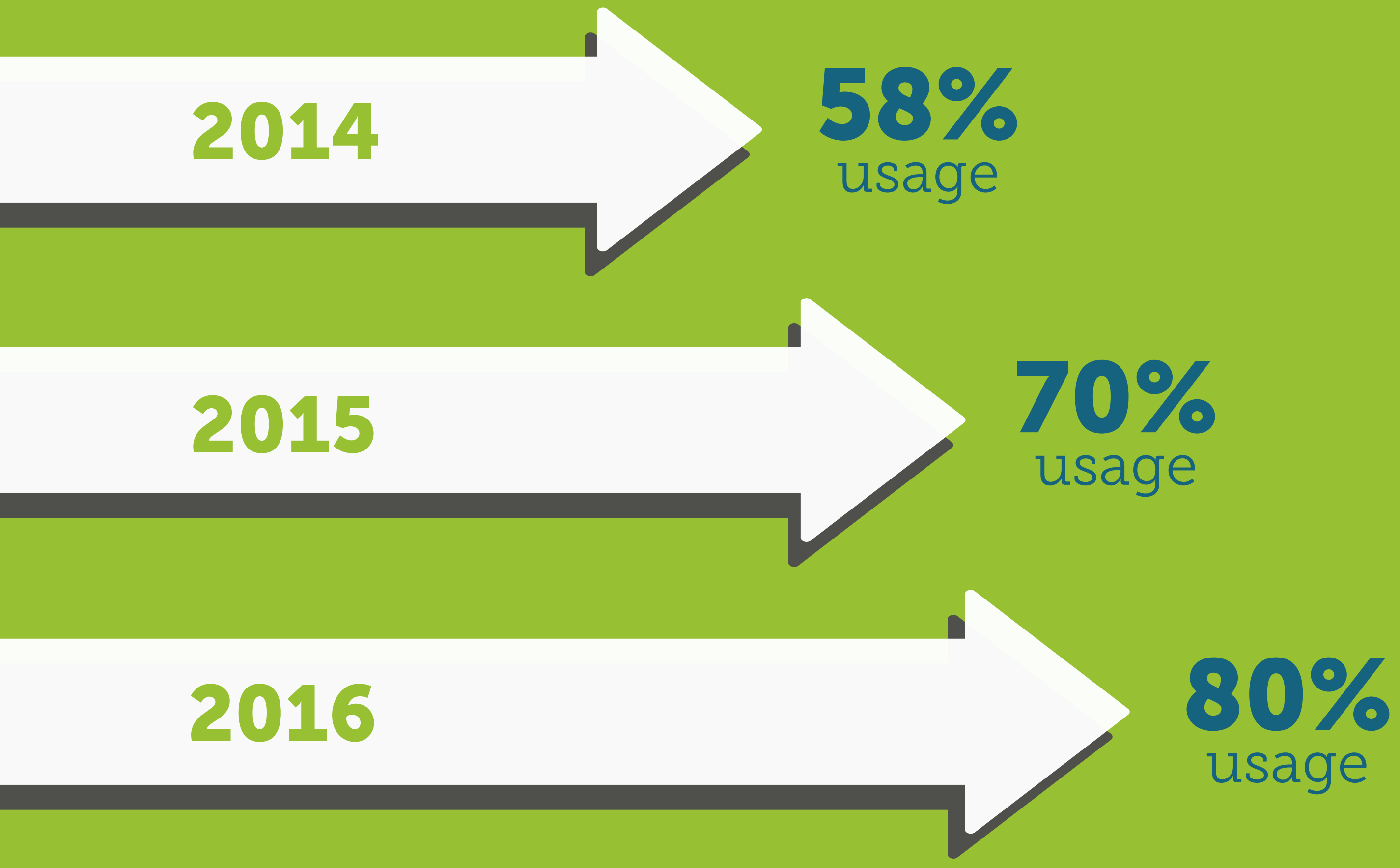


The Cool Kids Use Automation, Microservices, & Containers

Automation tools most frequently mentioned in order:

1. Puppet
2. Chef
3. Ansible
4. SaltStack

Automation tool usage is still rising.



2014 — 58% usage

2015 — 70% usage

2016 — 80% usage

Microservices and containers:

70% of respondents are moving to microservices and containers.

In their words...

"We are currently migrating to containers in production and to microservices for small parts of the backend infrastructure in logical domain groupings."

"Our main application is currently large, and has too much linkage between domains that aren't necessarily related. We're slowly beginning the process of identifying areas that can and should be split out into microservices, for ease of development and organization."

Chat is One of the Top-Cited Remediation Tools

Most responders state that their organization is at the beginning or intermediate stage of ChatOps adoption.

89% report using a group chat tool.

80% of respondents report that they are doing some form of ChatOps.

41% use a Chatbot. Hubot is most common, followed by homegrown versions.

Learn more with Jason Hand's O'Reilly book: Managing Operations with Group Chat

The Chat Platform is the Second-Most Important Tool During a Firefight

The most important tools during a firefight, in order, are:

1 Logs

2 Chat Platform

3 Monitoring Tool Dashboard

4 Graphite or Graph Tools

5 Runbooks

6 Email

In their words...

> “Chat and ChatOps let us handle multiple streams of conversation and also frees devs to work toward a solution without having to listen to chatter on a conference bridge.”

Learn more with Jason Hand’s O’Reilly book: Managing Operations with Group Chat

Post-Mortems Have Good Intentions

Most respondents have a healthy post-mortem process, most commonly focused on learning and improving and explaining outages.

- Blameless post-mortems continue to trend upward. Only **7%** stated that the purpose of post-mortems is to assign blame.

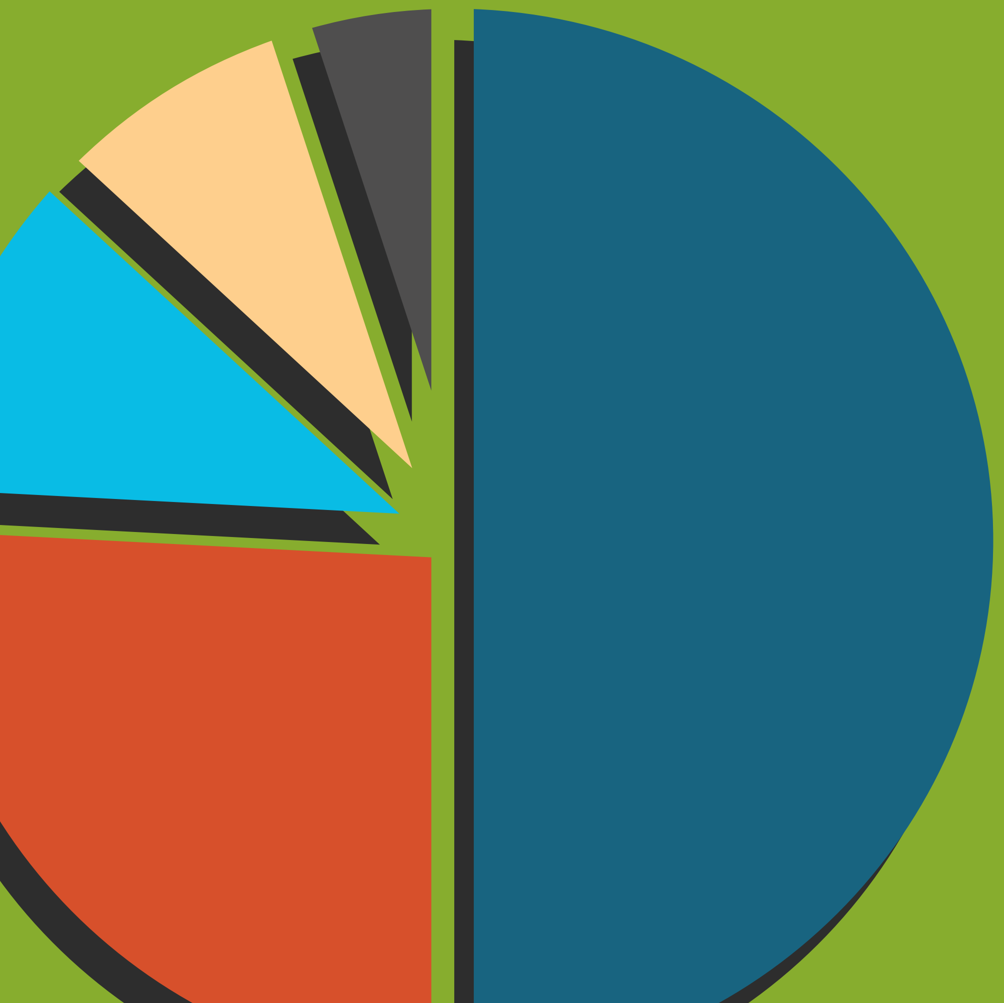
Biggest benefits of conducting post-mortems were (in order):

1. We gain a feeling of empathy across different departments.
2. We uncover bottlenecks and areas of friction in our processes.
3. We are able to update remediation information more quickly.

Download the guide to Blameless Post-Mortems

ITIL Proves Too Restrictive

Organizations that are more DevOps-focused find ITIL to be restrictive and antiquated. Others use parts of ITIL successfully.



- 🟦 **50%** have never used ITIL
- 🟥 **26%** use a combination of ITIL and DevOps/Agile practices
- 🟦 **11%** don’t know what ITIL is
- 🟧 **8%** have ITIL as the standard at their organization
- ⬛ **5%** are transitioning away from ITIL

In their words...

“ITIL is far too structured, and doesn’t meet the operational needs of a modern, fast-moving enterprise.”

“Structure is too rigid, the organization would like to approach alternatives but it is mandated by regulations and customers.”

“ITIL has an appropriate time and place, if you are delivering well defined services, then ITIL or six sigma is your best bet, but if you’re developing something that is not well understood, you’re better served by an Agile methodology.”

“As usual with methodologies, we only take from ITIL what’s matching our needs. Some principals are useful, others aren’t. They are complemented by some kind of Agile.”

Loss of Revenue Hit 56% of Respondents

Of those experiencing the costs of downtime, revenue loss hurts the most.

Others, by frequency of mention, include:

(47%) Diminished Perception Of Brand

(39%) Customer Defection To Competitors

(31%) Negative Publicity

(6%) Decreased Stock Price

(5%) Competitor Retaliation

Others costs mentioned:

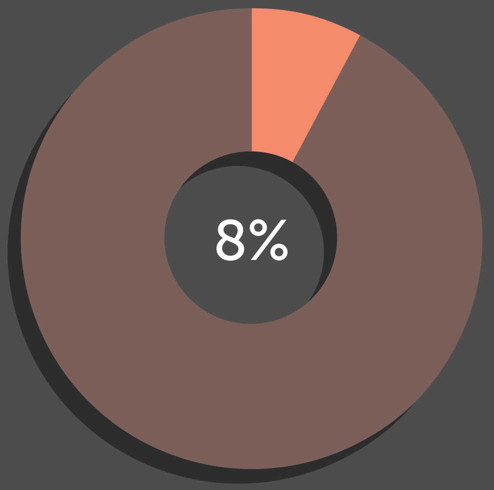
Bad demos, grumpy employees, poor morale, refund credit

The costs of downtime can be steep. Get the Guide to Real-time Incident Management to make the case for investing in processes and solutions to minimize its impact.

Incident Management Maturity has Room to Improve

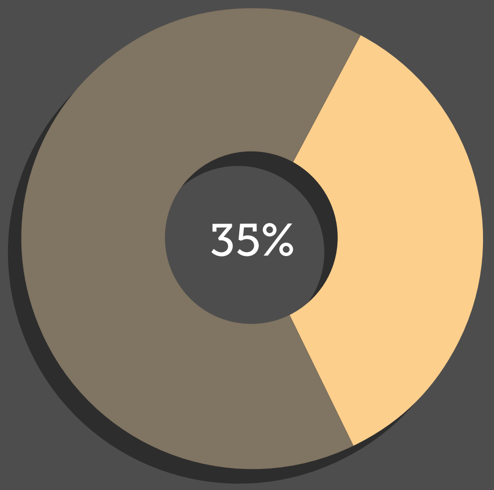
Most respondents ranked in the middle of the incident management maturity scale, meaning their orgs have implemented key health indicators and real-time collaboration processes.

What is your Incident Management Maturity?



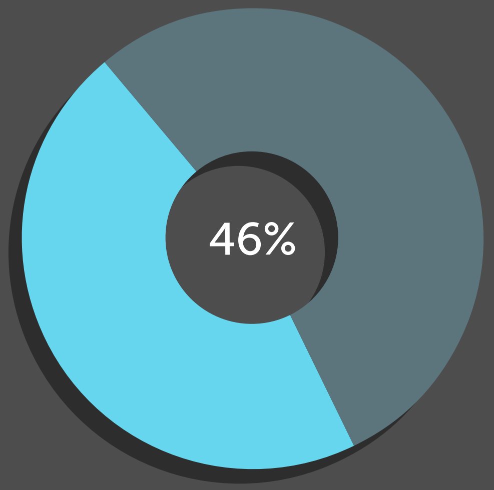
Beginner

We have little-to-no operational stack awareness and poor communication between Dev and Ops.



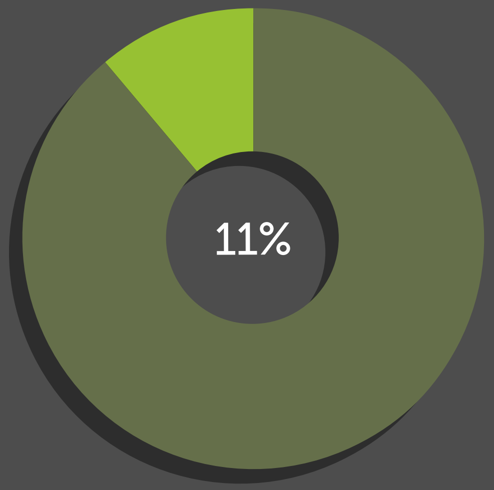
Novice

We have some defined processes and roles, and some monitoring.



Intermediate

We have consistent real-time collaboration processes, and key health indicators.



Advanced

We have full stack awareness via automation and proactive empathy across the delivery cycle.

Learn how DevOps plays into the Incident Lifecycle

Predictions for the Future

- **More people will take on-call responsibilities, covering their area of expertise.** An “all in this together,” culture of uptime mentality, coupled with cradle-to-grave code ownership, will lead to more experts on-call, fewer handoffs, and faster incident resolution.

- **Rapid recovery will be increasingly critical for company viability.** The expectations and demands of today’s consumers mean that failing to continuously deploy and recover quickly from inevitable hiccups could lead to obsolescence.

- **The pure operations role continues to fall out of favor.** As automation, microservices, and containers continue to replace heavyweight legacy systems, developers will take even more ownership and authority over their code. The purely operational role will fall to the wayside.

- **“Alerting” as table stakes will decline in favor of real-time incident management.** Simple alerting systems have proven noisy, lack context, and fail to deliver. Real-time incident management, with its focus on signal, continuous improvement, and reduction in Mean Time To Resolution (MTTR), will become de rigueur.

- **Downtime will become even more expensive.** More organizations will quantify their costs of downtime rather than just know it’s expensive. Investment in incident management will sharply increase in response to ever-increasing demands for uptime or at the very least, rapid recovery.

THANK YOU

DevOps.com

O'REILLY®

FOUNDRY GROUP

SRE WEEKLY
scalability, availability, incident response, automation

VividCortex

DEVOPS WEEKLY

New Relic®

If you need help improving your on-call process, we've got a trial to show you just how we're making the process suck less.

SEE HOW HERE

THANK YOU

If you need help improving your on-call process, we've got a trial to show you just how we're making the process suck less.

victorops.com/start-free-trial/