

# 2020 SRE Report

THE DISTRIBUTED SRE





## SRE Survey Contributors

Catchpoint would like to extend a special thank you to Sanjeev Sharma, Marc Hornbeek, Archana Joshi, and Niladri Choudhuri. Their insights and contributions set the trajectory for this entire report.

We would also like to extend a special thank you to Nithyanand Mehta. Nith's internal white paper on maturity was in inspiration for some of this report's key talking points.

Thank you to Eveline Oehrlich and colleagues at the DevOps Institute. Their feedback and time were critical contributions to more than can be captured in this document.

## Supporting Partners

Catchpoint could not have run our SRE from Home Survey without our amazing partners Blameless, Gremlin, Honeycomb, NS1, Launch Darkly, and Packet.

# Foreword

The vicious cycle of rising customer expectations has been long discussed as a driver for increasing complexity of delivering services in a fast, edge-distributed, and reliable way. This year, we give special consideration to a sudden increase in work from home; we see our workforce as distributed as our customers.

We are truly grateful to offer the 2020 SRE report with two clear data sets. This year's report includes survey results and data from both "pre" and "post" work from home periods of time, offering one of the industry's most unique perspectives on what it means to be an SRE in 2020.

We evaluated the data from over 600 hundred survey respondents. As we analyzed the data, we hoped to create an honest, humane look at the trends, status, and challenges facing today's SRE pioneers.

As we offer our heartfelt thanks to all individuals who contributed to this report, we now offer that same thanks to you, the reader. We hope you enjoy reading as much as we enjoyed researching and writing.

Like previous SRE reports from Catchpoint, data was considered from individuals who identified as doing SRE-type work, even though the SRE title may not have been used.





# Table of Contents

Sponsors	1
Foreword	2
Introduction	4
Key Takeaway 1 <b>Observability Components Exist; Observability Does Not</b>	5
Key Takeaway 2 <b>Heavy Ops Work Load Comes at a Cost</b>	13
Key Takeaway 3 <b>Shift to Remote Creates Opportunities and Challenges</b>	22
Key Takeaway 4 <b>The Future of SRE is Remote and Bright</b>	32
Methodology	41



## Introduction

Starting with the question, “What happens when you ask software engineers to design an operations team?” results in the answer, “SRE teams are responsible for the availability, latency, performance, efficiency, change management, monitoring, emergency response, and capacity planning of their services.” If SRE is a narrower implementation of larger DevOps principles, then the primary distinction is SRE’s core focus is on reliability.

Using the above question and answer as a line in the sand, this year’s SRE 2020 report highlights an objective which may be common among relevant practitioners, regardless of their title: designing observable systems to prevent service disruptions instead of reacting to them. It starts with a clearly-identified convergence point and works backward so big or small organizations alike can evaluate against this 2020 baseline.

If a common objective is to solve complex problems, then what does that journey look like? In a microservices world, driven by edge computing efforts, the journey involves more components than before, and these components now need to be re-evaluated in a work from home reality. This includes surfacing areas which may have been ignored or nonexistent. Consider things like morale, employee experience, and human wellness to go along with traditional asset classes like organization structure, tool stack, and hardware and software.

## KEY TAKEAWAY 1

# Observability Components Exist; Observability Does Not

Identify where your provided services converge into the quintessential [digital] experience point of consumption; work backward from there. Be sure to include consideration for not only your code, but also the networks, third parties, and all delivery chain components to evaluate how well the three observability pillars are applied through an experience lens. Ask, "Is the customer's experience the way it is because of code, the internet and networks, third parties, or other delivery chain components?".

Identify where your provided services converge into the quintessential [digital] experience point of consumption and work backward from there. Ask, "Can users reach our services from where they are?". If we offer that capabilities are the gateway to positive business outcomes, hardly one can argue being preventive through designing and building observable systems is a necessary capability in today's edge-distributed, experience-centric world.

When presented with the question of, "What tool categories are used by SREs?", a whopping 93% chose monitoring compared with 53% choosing observability. When we dug into further indicator questions, a bright, shining light challenged and invited us to take a deeper look at some of the monitoring entrenchments.



If observability's academic definition is, "How well internal states of a system can be inferred from knowledge of its external outputs", then we must attach to a contextual definition of the experience, usually that of a consumer, customer, or an employee, when we say outputs.

**Consider:**

*If a user's digital experience consists of third parties, networks [internet and internal], code, and infrastructure all converging at a critical point in which they become what we refer to as an experience*

*Then do not let observability's commercial definition of events, metrics, and tracing place a disproportionate amount of focus on white box internals*

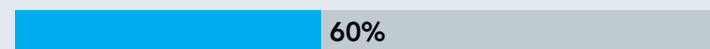
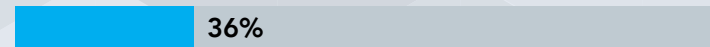
# 71%

**of respondents cited error rate as a key metric they track.**

Seventy-one percent of respondents cited error rate as a key metric they track. Stating customer satisfaction (see next section for data) is a high priority but measuring error rate instead

of end user response time is causing continuous focus on looking from the inside out instead of from the outside in. Rather than debate various white box versus black box monitoring theory, instead focus on understanding the correlation between the experience and the components going into the delivery of the experience.

## Which of the following metrics are tracked by your organization?

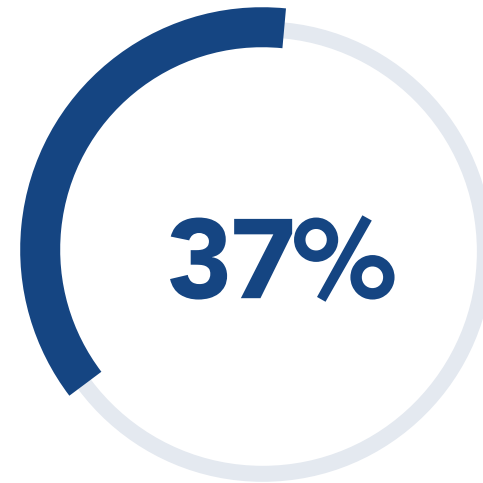
**Error Rate****End User Response Time****MTTR****MTTD****Error / Performance Budgets**

What also warrants a discussion is the prolific lack of attention or visibility to third parties. According to HTTP Archive, 93% of pages include at least one third-party domain; an average page includes nine unique third-party domains! Yet only 11% of respondents said their automated workflows extended to third-party providers.

Given that observability's pillars must also apply to third-party components, it may be understandable of why there is a gravity to focus on only white box internals. Just as SRE working to design observable systems is relatively new, using digital experience monitoring to shed light into third-party systems and collect data is also relatively new. Here, though, lies a golden opportunity to consider the extension of black box digital experience monitoring to include third parties. Relying solely upon white box monitoring means you are not aware of what the users see, especially as it pertains to third parties. For example, pages that do not load or apps that do not navigate may be the result of a misbehaving CDN, transit network, or DNS provider.

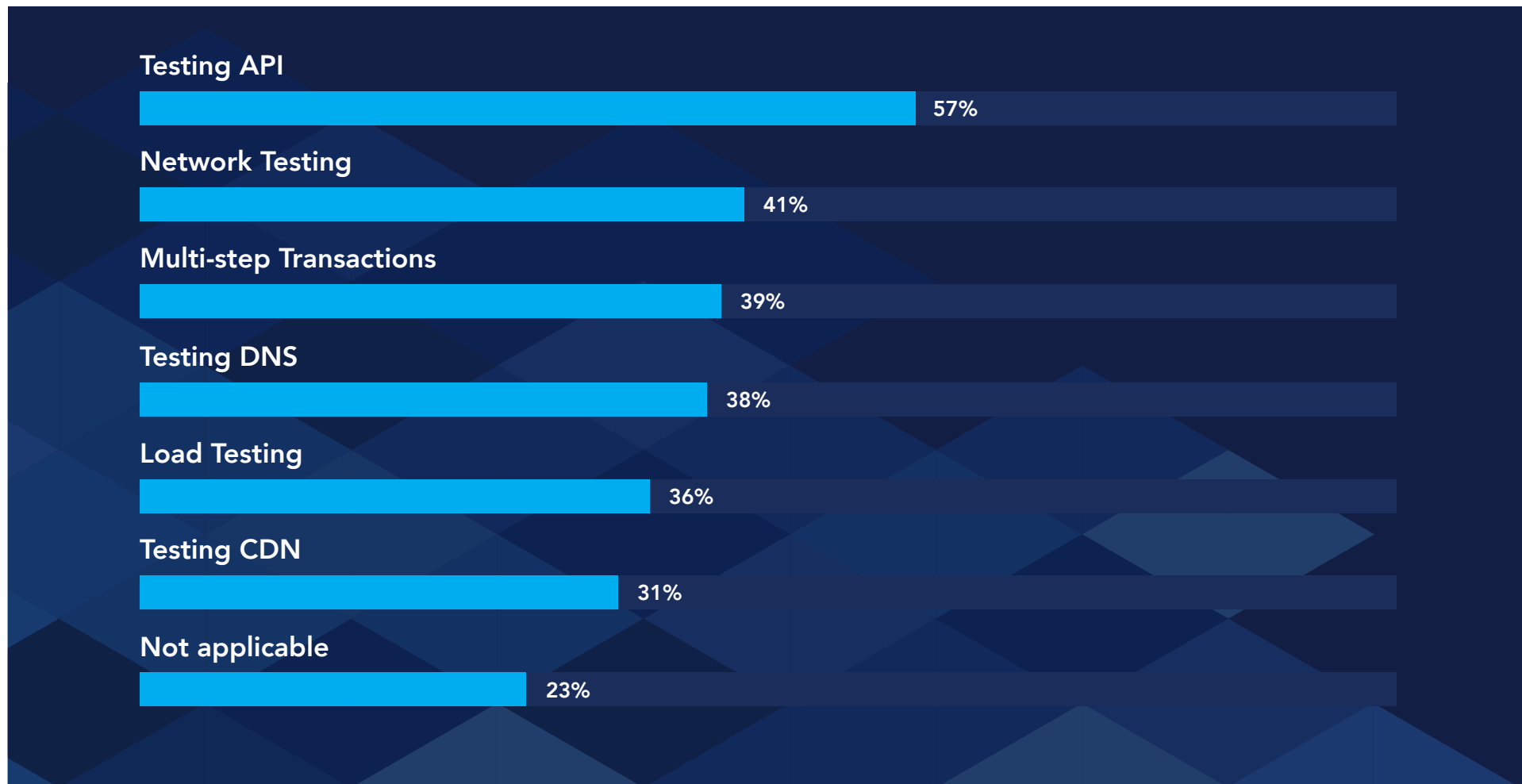


**11% of respondents said automated workflows for incident management include all third-party providers**



**37% of respondents cited third parties as the cause for increased incidents (second to only traffic/capacity issues) while at home**

## Which of the following strategies are used for synthetic monitoring?

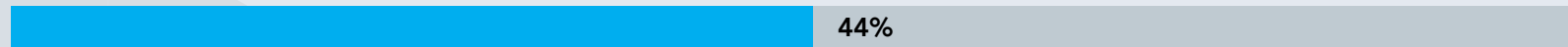


Of the synthetic users, a key indicator here is only 39% were using synthetics for multi-step transactions to emulate an experience.

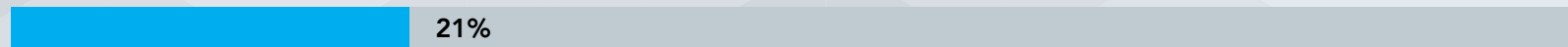
Compare with other use cases which monitor a specific component (e.g. DNS or CDN), or with respondents who do not use any synthetic monitoring at all.

## To what extent has the SRE Team implemented comprehensive application and infrastructure performance monitoring and alerting?

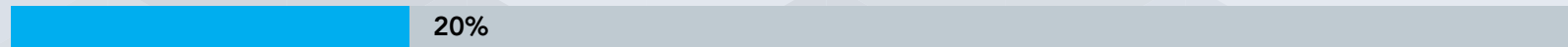
Monitoring and alerting are highly automated. Monitoring systems are intelligent enough to discern need to alert for specific events only



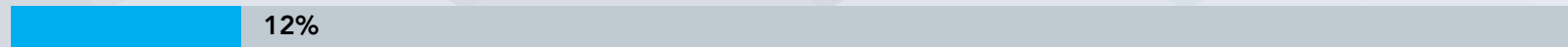
We have system-level performance monitoring and automated alerting, but not at the application level



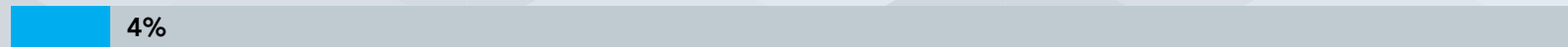
We have visibility into some application and systems but there are key areas where we have no capability to monitor



There are just dashboards for real-time monitoring. There is no automation



There is no real-time monitoring. The on-call team gets manual alerts from customer support



Eighty-nine percent of respondents said they perform monitoring activities, with 44% saying monitoring and alerting are highly automated. This is good news for indicating white box internals are well considered. The bad news, though, is a clear focus on the inside-out causing us to theorize that outside-in black box monitoring, focusing on the digital experience, is still misunderstood. For this, we offer the following perspective for companies to evaluate as they mature to preventive measures through designing observable systems:

***Identify where your provided services converge into the quintessential [digital] experience point of consumption; work backward from there. Be sure to include consideration for not only your code, but also the networks, third parties, and all delivery chain components to evaluate how well the three observability pillars are applied through an experience lens. Ask, "Is the customer's experience the way it is because of code, the internet, third parties, or other delivery chain components?"***

**89%**

perform monitoring activities

**44%**

said monitoring and alerting were highly automated

**21%**

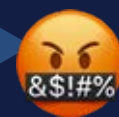
said monitoring and alerting were automated, but only at the system level

**11%**

said automated incident management included third parties

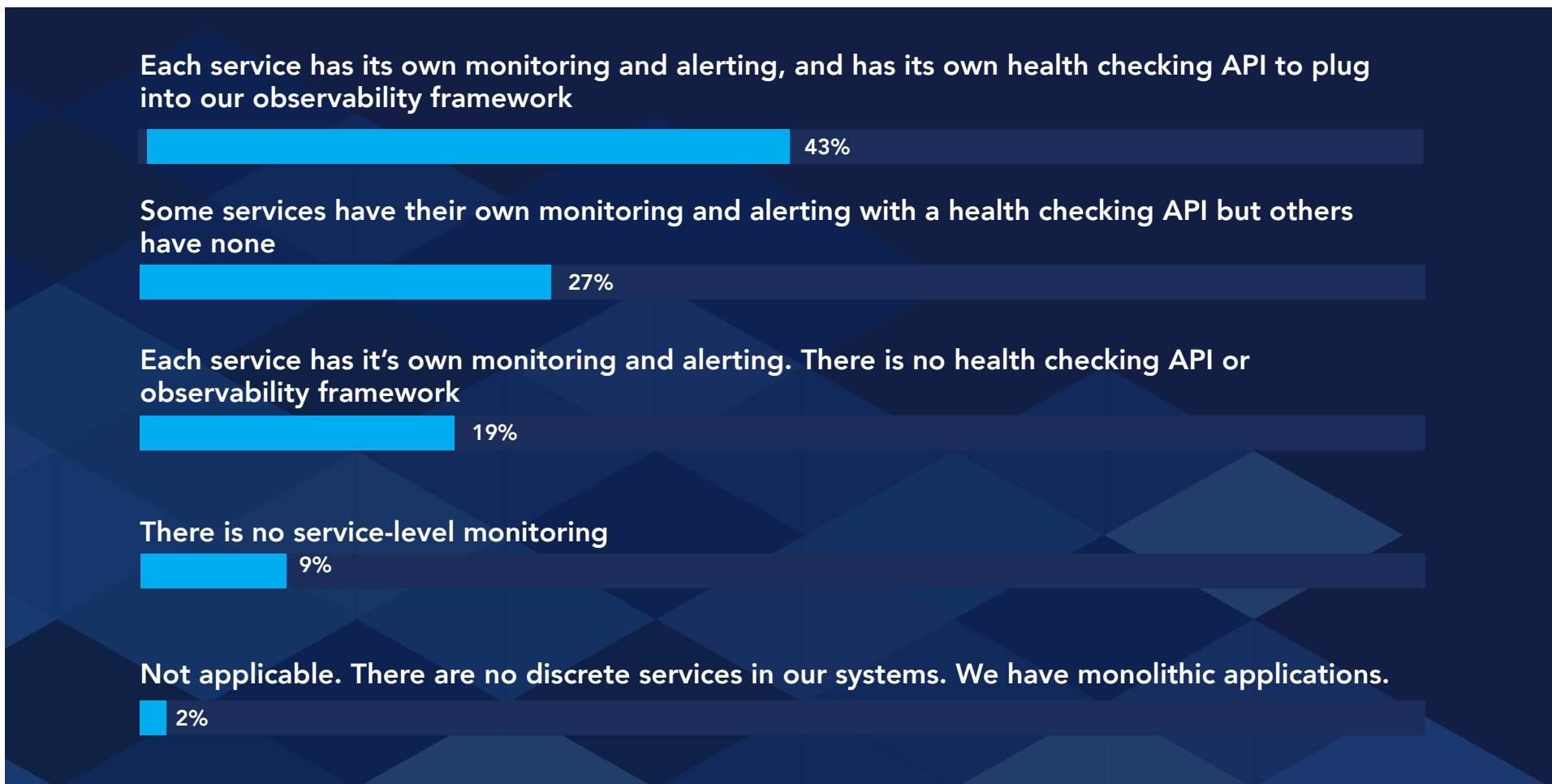
**11%**

said there is no monitoring at the service level of any kind

**4%**

said there is no real-time monitoring and they receive notification calls from customer support

## Is there health monitoring at the service level to be able to detect outages or performance issues (at the service level)?



Observability is about answering previously unanswerable questions as it pertains to “why”. “Why” can’t users reach my site? “Why” can’t users access their data? “Why” is the user sentiment as low as it is?

The ability to answer “why” should be powered by a framework, and not an individual tool. This is such an important indicator question, which we offer to close out this section. If 43% of respondents plug their data into an Observability framework, then 57% do not. In the next section, we explore this gap further by looking at some of the key “Dev” versus “Ops” data.

## KEY TAKEAWAY 2

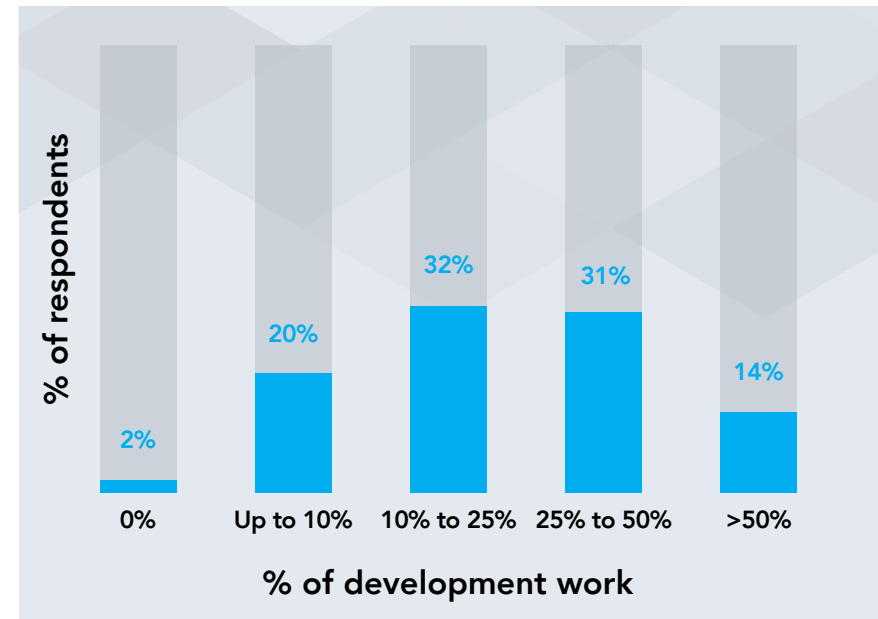
# Heavy Ops Work Load Comes at a Cost

Implement DevOps' SRE principles to prevent incidents by designing and building observable systems. Work to shift reliability further left, offering the benefits of reduced cost, team alignment, and business outcomes. Use the 50/50 dev work versus ops work split as a guideline, with no more than 25% of ops work being on call. Then, as you contextually iterate toward the preventive end goal, identify constraints to remove them. Capture results to form the basis of a charter. As you remove constraints, then update your charter accordingly.

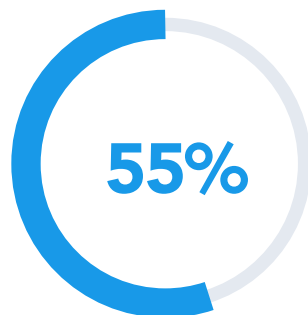
If up to 90% of the cost of owning a system is after its deployment (i.e. shifted right), then why do businesses still approach in a predominantly ops-type, reactive way? In this key takeaway section, we explore this question and offer that businesses' SREs have an opportunity to shift left to help take all their work and transform into a mature, observability capability.

Google suggests there should be an upper bound goal of 50% ops work and 50% dev work (conversationally referred to as the, "50/50 split"). Ideally, the amount of ops work should be much less than this. Part of the ops work should be no more than 25% of on call. The goal of having a 50/50 workload split between doing dev activities versus ops activities seems to be a pipe dream. According to survey data, most of the work is dominated by operations-type activities.

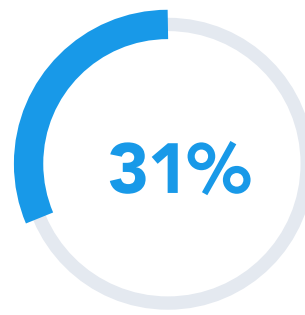
## What percent of SRE time includes working on development work?



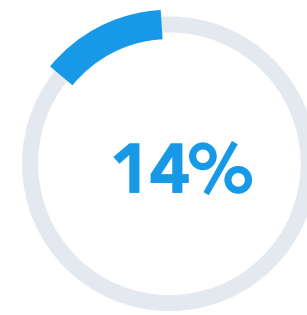
When asked the question of, "What percent of your work is spent on development?", only 14% said more than 50%:



said between 0-25%

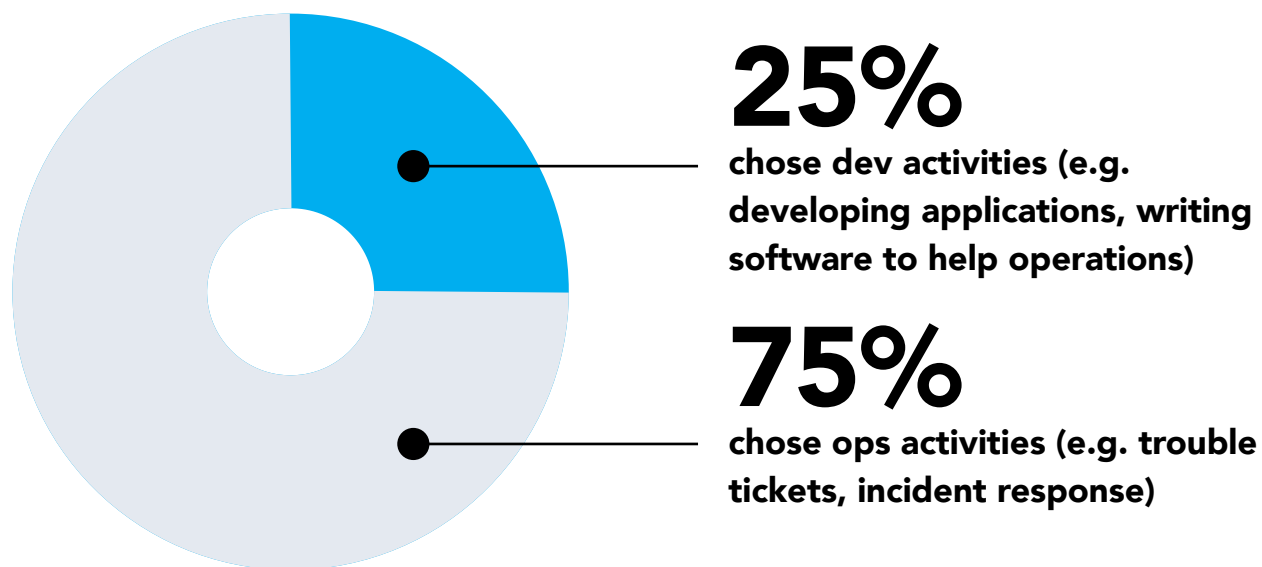


said between 26-50%



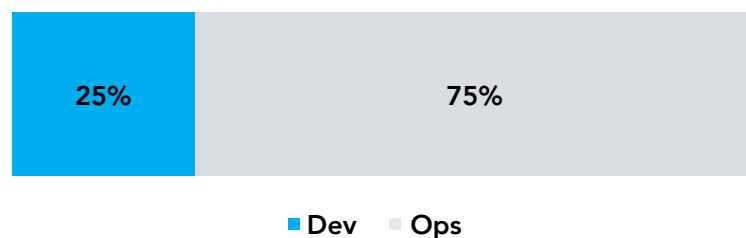
14% said greater than 50%

When asked essentially the same question (but with listing specific choices for people to select) as, "Which of these activities do SREs do as part of their job?" an eye-opening:

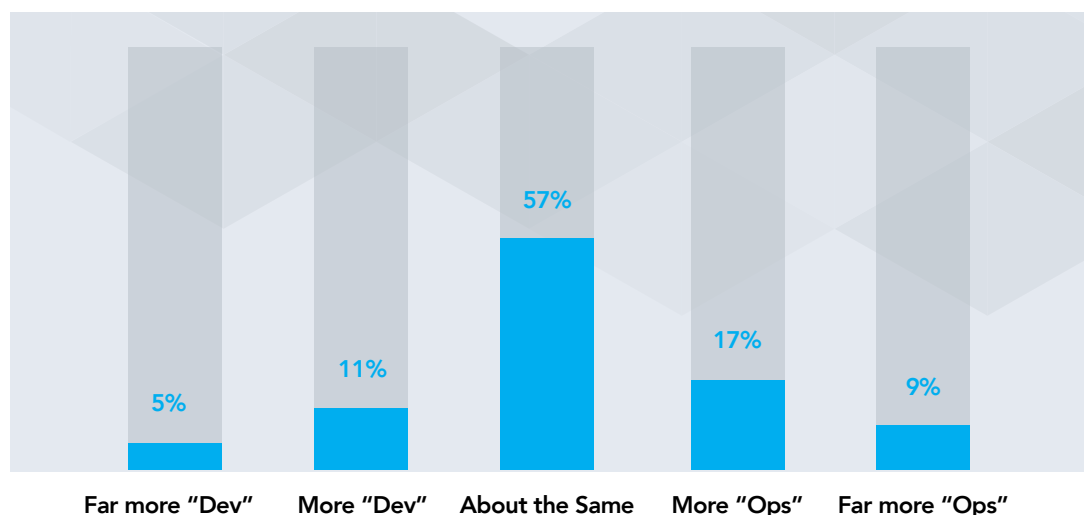


After two and a half months of work from home, a net 10% of survey respondents said their activities have shifted to include more ops work.

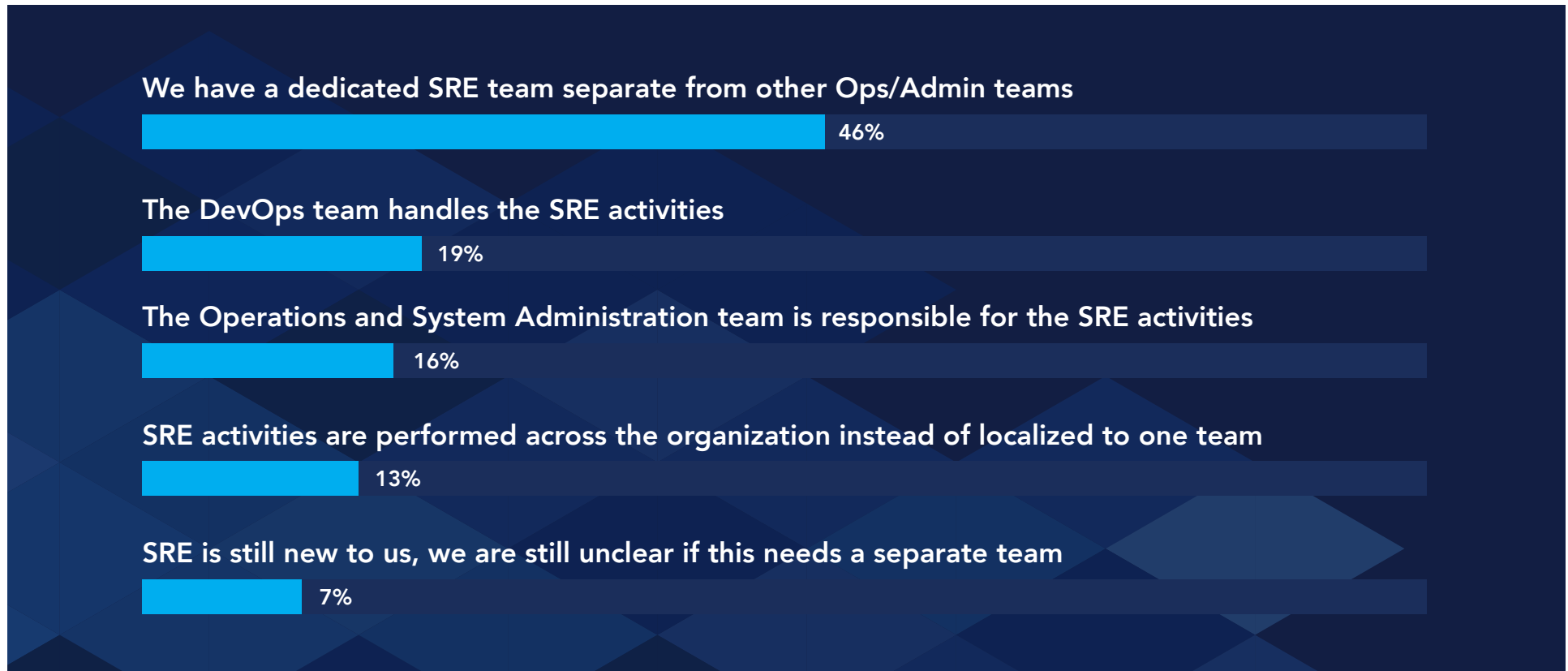
Dev vs Ops Work Distribution



How have your activities shifted since at home?  
(Dev vs Ops)



## Who performs SRE activities in your organization?



If we are all on a journey to being preventive through the design and implementation of observable systems, then we have a long way to go. First and foremost, consider a build it and they will come approach to transforming an SRE organization. Start by recognizing the work being identified as DevOps' SRE. According to our survey, 83% identified as doing SRE activities. We caution, though, identifying as doing SRE activities does not mean being an SRE. This is because we must consider as a whole, not as parts or pieces. SRE teams are becoming more defined but spanning across different focal points does lend SRE work to being buried or hidden.

Forty-six percent claimed there is a dedicated SRE team. However, 53% said they were challenged by being involved late in the lifecycle and 52% said they spent too much time debugging (more on this later): key anti-SRE indicators.

Reacting to incidents and problems is a part of the SRE life. If we re-introduce a core goal of being preventive through the design of Observable systems, then the stages of the journey might look like this:



In that vein, we asked what reactive activities do SREs perform. The purpose here was to help identify where companies may start to mature from being reactive to proactive, based on their business and organization's context.

Looking at each line item result, postmortem analysis and respond to system-generated alerts were one and two, respectively. However, another way for readers to look at these results is to group some of the responses into categories, and then decide if you should work to mature a given category versus a given line item. For example, if the type of analysis on postmortem has overlap with the type of analysis for metrics including SLI and SLO, then consider whether overall analysis can be a place to start on the path to preventive.

## Which “reactive” activities do SREs practice within your organization?

Postmortem analysis of problems via planned activities



Respond to system-generated alert messages



Analysis of metrics including SLIs, SLOs, SLAs



Document knowledge gained



Repairs to infrastructure problems



On-call rotation



Review and respond to support tickets reported by clients



General administration tasks (e.g. progress reports, housekeeping)



Reproduce problems reported by clients



Install, configure, and/or commission applications for clients



If one is not reacting to a fire, then we may consider everything we do is proactive. Rather than debating the nature of an activity in isolation, instead shift the conversation and attach to “what are we preventing from happening by doing this?” This way, the conversation can shift to a results-based approach when discussing what an SRE charter will look like. Ideally, we want to optimize service operations to a level where ongoing human work is no longer needed, so the SRE team can focus on high-value engagements.

## On which “proactive” activities do you spend a moderate or large amount of time?

Automate tasks so that they need not be performed manually



Monitor and analyze system metrics for trends that could result in failures or SLA issues in the future



Support post-deployment operations



SRE-specific systems planning



Write software to help operations



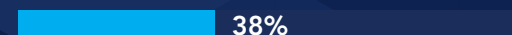
Work with development to help develop applications



Preventive maintenance of systems



Capacity planning activities



Resiliency checks via practices like chaos engineering



When removing either the proactive or the reactive qualifier and asking, “Which of these activities do SREs do as part of the SRE job”, we see monitoring and incidents are cited as top activities. What is also important is development to help with applications is ranked fifth. Keeping in mind that dev should be the predominant activity type (versus ops), consider what has caused this to be the current state:

- ✓ **Are charters incorrect or non-existent?**
- ✓ **Have SRE principles not been considered as necessary?**
- ✓ **Has a customer-centric approach to delivering services not been evaluated?**
- ✓ **What else can be asked? And “why”?**

## Which of these activities do SREs do as part of the SRE job?



Once SRE work and value becomes recognized, then it can start to be rewarded. To help garner support, attach the conversation to some type of business context. For example, reducing ownership costs of systems occurs when reliability is considered earlier than later. It is much easier to make a reliable system, more reliable.

Focus the conversation on the drive and desire to solve complex problems and achieve business goals; consider this datapoint as a baseline:

41% said DevOps and SRE are part of the same team  
29% said DevOps and SRE are complementary  
19% said I do not know  
11% said they are competitors

## How do you consider the SRE and DevOps team relationship?

Part of the same team

41%

Complementary

26%

I don't know

19%

Competitive

11%

## How important are each of these metrics in terms of measuring the business impact of changes?

Drop in customer satisfaction

82%

Lost revenue

79%

Lost customer

79%

Drop in employee productivity

69%

Social media backlash

59%

Fortunately, survey respondents were able to articulate how success was measured in terms of the business.

Capabilities are the gateway to positive business outcomes. When having the, "here's why we need SRE" conversation, do not talk about only capabilities or only positive outcomes. Instead, combine them by saying, "these capabilities will help us achieve these outcomes".

To close this section, there is a huge opportunity to shift left, from reactive operational work, into earlier stages. From “not just helping with development”, but “taking feedback from outputs of work and using as an input to help in aiding the observability of offered products or services”. For example, after an SRE picks up that pager, their documented dos and don’ts may lead development on the next service to avoid pitfalls.

Our last thought is the idea of SRE engaging in the overall lifecycle applies regardless of the size of the organization. In other words, the stages of the journey are still the same.

***Implement DevOps’ SRE principles to prevent incidents by designing and building observable systems. Work to shift reliability further left, offering the benefits of reduced cost, team alignment, and business outcomes. Use the 50/50 dev work versus ops work split as a guideline, with no more than 25% of ops work being on call. Then, as you contextually iterate toward the preventive end goal, identify constraints to remove them. Capture results to form the basis of a charter. As you remove constraints, then update your charter accordingly.***

## How many dedicated teammates are involved in SRE activities?

0-5



6-10



11-20



21-30



31-40



41-50



More than 50



### KEY TAKEAWAY 3

# Shift to Remote Creates Opportunities and Challenges

Turn newly-surfaced, or previously-ignored, challenges into opportunities for strategically differentiating. Focusing on challenges like morale, employee experience, work/life balance, and employee engagement & sentiment may showcase a company's employee-first mentality to attract or retain top talent.

**An SRE by any other name would still add business value. But does the business place value on their SREs?**



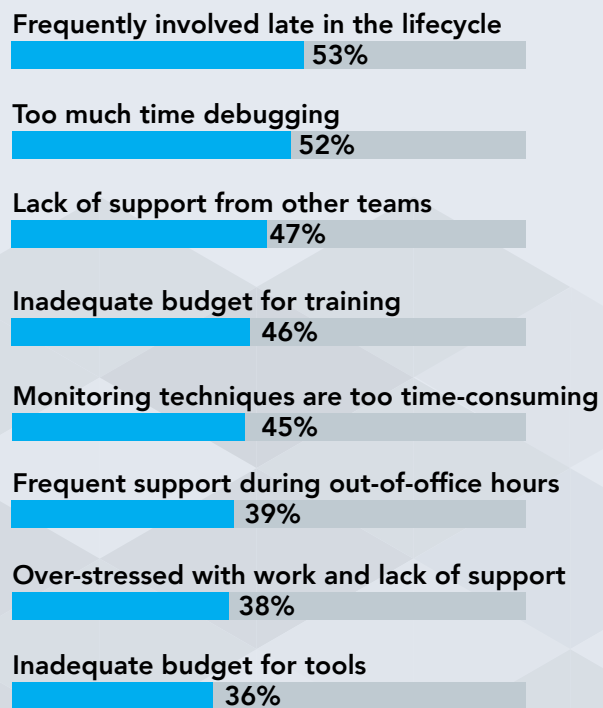
**If, “treat your employees right and they will treat your customers right” was just a motto, then may it now be your battle cry.**

In the pre-work from home set of SRE 2020 questions, respondents identified some key challenges they were facing. Then, after two and a half months of work from home, additional sets of challenges surfaced.

This includes the surfacing of challenges which may have been previously ignored, or non-existent to some. Challenges like morale, employee experience, work/life balance, employee engagement, and sentiment now have an opportunity to be turned into strategic, differentiating assets which showcase a company's employee-first mentality. In other words, if “treat your employees right and they will treat your customers right” was just a motto before, then may it now be a battle cry.

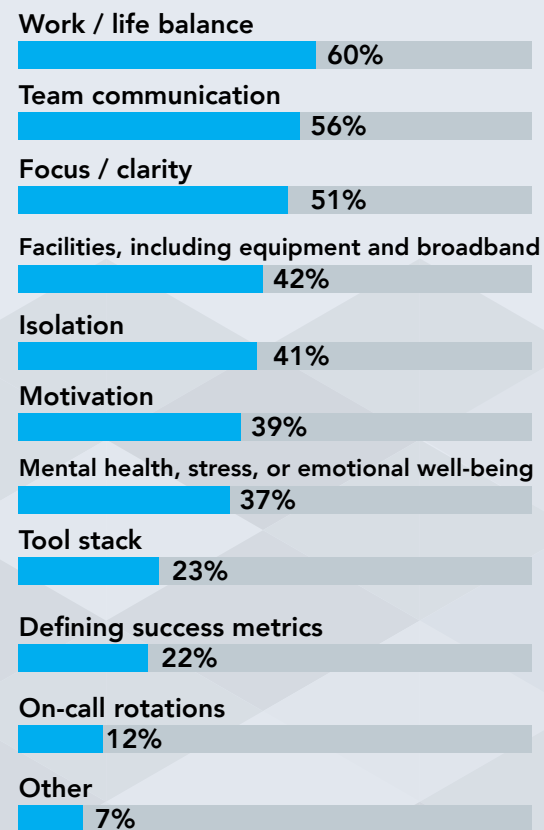
## PRE

What issues are somewhat or extremely challenging?



## POST

What issues are you facing since 'at home'?



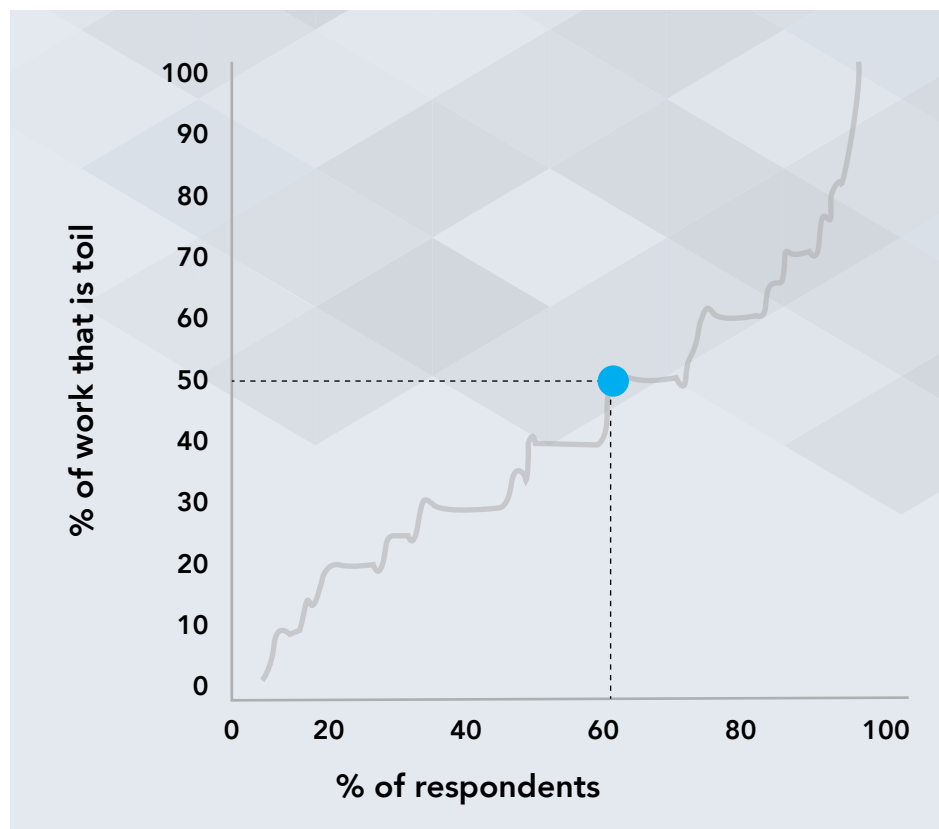


**"I have found having my kid at home with me everyday to be the most stressful part of this experience. Maintaining a work-life balance can be tough in general, but when you are losing focus a lot, it can be hard to not feel like you are working as hard as other people (even when your company is supportive)."**

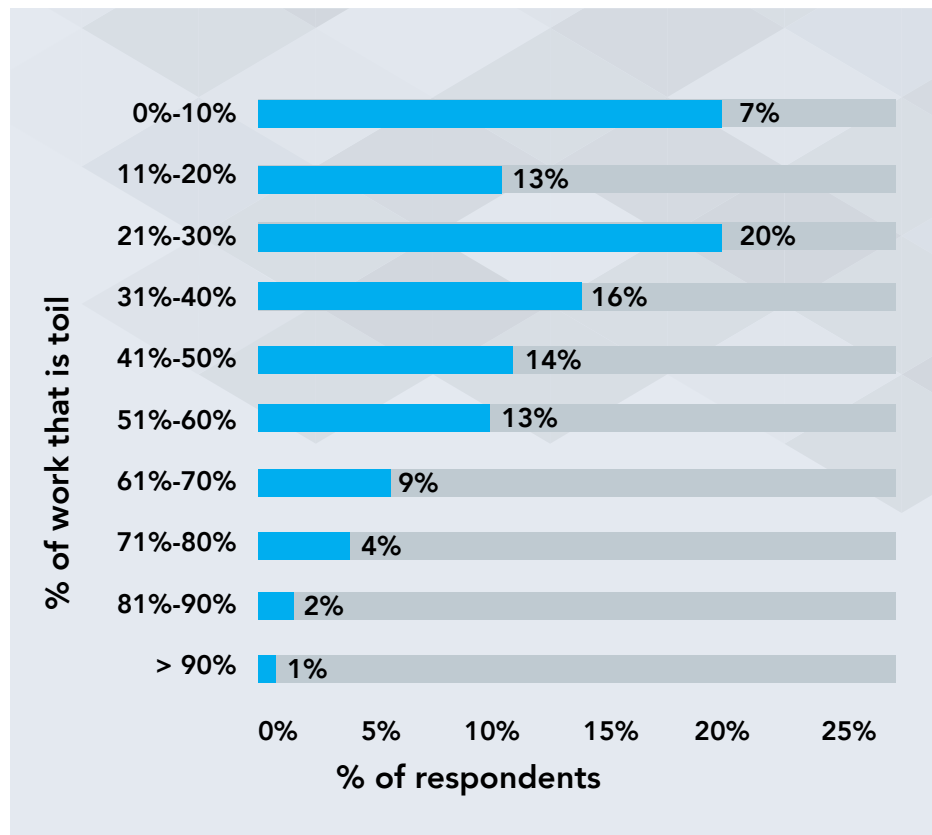
**— SURVEY RESPONDENT**

In our 2019 SRE report, prolific toil sprawl was front and center with 59% of respondents believing there is too much toil in the organization. In our SRE 2020 report, we revalidate this finding and offer an expanded, distributed look at the data:

## What percent of SRE work is toil?



## What percent of SRE work is toil?



Forty-one percent of respondents said half, or more, of their work was toil. Considering both 1) high rates of toil and 2) the addendum question which had 60% of respondents listing work/life balance as the number one challenge since work from home, we suggest that businesses take a tactical look at approaches to reduce burnout.

Here, we make a distinction from toil (which in and of itself may be just the type of mental activator someone needs from time to time) versus burnout, which is the result we are trying to avoid.

When dealing with the high amounts of toil in an organization, consider the underlying reasons for the lack of automation. If the conclusion is the result of a lack of skillset or aptitude, then the path forward may be different than e.g. if a tremendous amount of technical debt has been accrued over a large period of time.

Could automated capabilities be scaled if teams were aligned on a common set of objectives, or aligned priorities? Is there a fundamental miss of combining dev + ops in first place? At a minimum, baseline where one stands regarding the 50/50 “dev” versus “ops” split to get an idea of the toil gap. Then, manage the conversation so teams have a reasonable expectation and do not, for example, feel like they should be doing zero ops work.

## What is the primary source of toil for SREs?

Manual maintenance tasks that could be automated



Work on application releases that could be automated



On-call manual tasks that could be automated



Resolving false positives/negatives



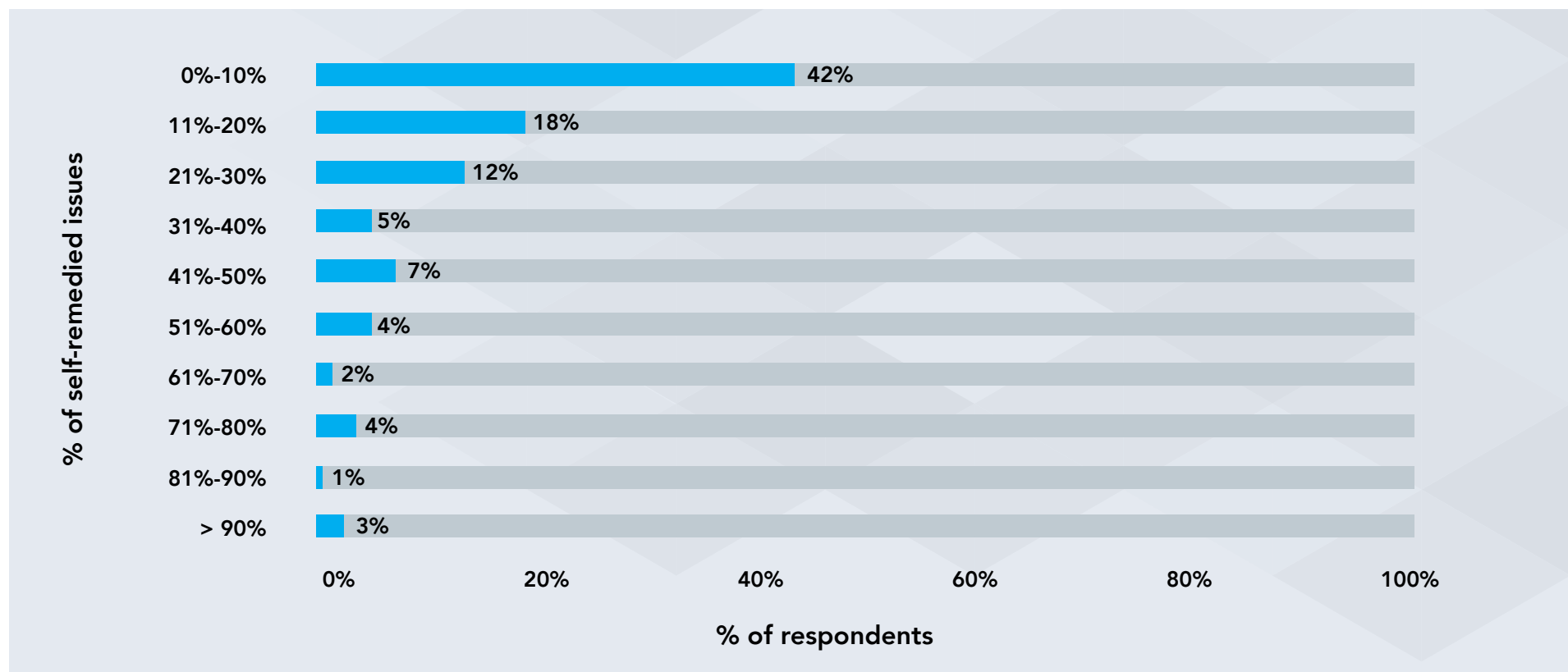
Non-urgent service related messages



Resolving non-service related messages



## What percent of issues and incidents are self-remedied using automation?



Forty-five percent said monitoring techniques are too time-consuming. Here is an opportunity to focus on preventive measures through an observability capability, while also expanding the use of software (instead of humans) to interpret data and whether an action is needed. Instead of generating an alert and asking a human to decide whether they need to take an action, generate alerts only if a human should take an action. Then have the system actually perform the action.

To help with this shift to actionable alerting, we say, “You can’t monitor and alert on everything, so start by monitoring and alerting on the most important thing: The Experience”. In this vein, various artificial intelligence (“AI”) or machine learning (“ML”) capabilities may be of use.

Sixteen percent cited resolving false positives/negatives was a primary source of toil.

We were a bit surprised to see this data. In our 2018 SRE Report, 65% of respondents said they were either wholly, or partially, in the cloud. We expect this number to have increased since then (although we did not explicitly ask this question this year). With various cloud provider capabilities, including various “features/functions as-a-service”, our advice for evaluating what is automatable to reduce toil and subsequent burnout remains the same: When considering the high amounts of toil in an organization, consider the underlying reasons for the lack of addressing automation. If the conclusion is the result of a lack of skillset or aptitude, then the path forward may be different than e.g. if a tremendous amount of technical debt has been accrued over a large period of time.

## Do you use automated tools for capacity planning and provisioning?

Capacity planning is all manual.  
Provisioning is automated.



Provisioning is manual and capacity planning is automated



Capacity planning and provisioning are both manual



Yes. We have full automation for predictive capacity planning and provisioning



Our capacity planning is done with our budget cycles and if frozen with budgets



Lack of budget and training (from the previous list of challenges) coupled with high amounts of toil can lead to burnout. As noted in the data from our addendum questions, these problems are exacerbated with work/life balance (60%) and focus/clarity (51%) being the two highest well-being challenges since work from home.

There is a correlation between in-house mentoring and training programs (78%) with inadequate budget for SRE training. This suggests that in-house programs may not be as effective. The data also begs us to ask whether too much time debugging (52%) is because training is a gap here as well. Since in-house training is the predominant training approach, look at these programs' effectiveness. Are the training coaches or leads experts in the field? Is this a challenge? Is the desire to do in-house training a direct consequence of no budget? Keeping in mind the desire is to make employees more productive, then in-depth training and solid understanding of the roles to fill the SRE charter are essential to implement a roadmap for preventive measures through the design and implementation of observable systems.

We also include the lack of budget for tools in this same section on training, as lack of budget was a common theme between the two. Unfortunately, measuring drop in employee productivity was the second lowest metric in terms of business impact, so investing in training may be a place to start.

## What trainings and certifications do your SRE team members have?

### In-house mentoring and training programs



78%

### Cloud certification (e.g. AWS, GCP, Azure)



51%

### Tool vendor training or specific tools



40%

### 3rd party DevOps courses delivered in individual or group class settings



29%

### 3rd party SRE courses delivered in individual or group class settings



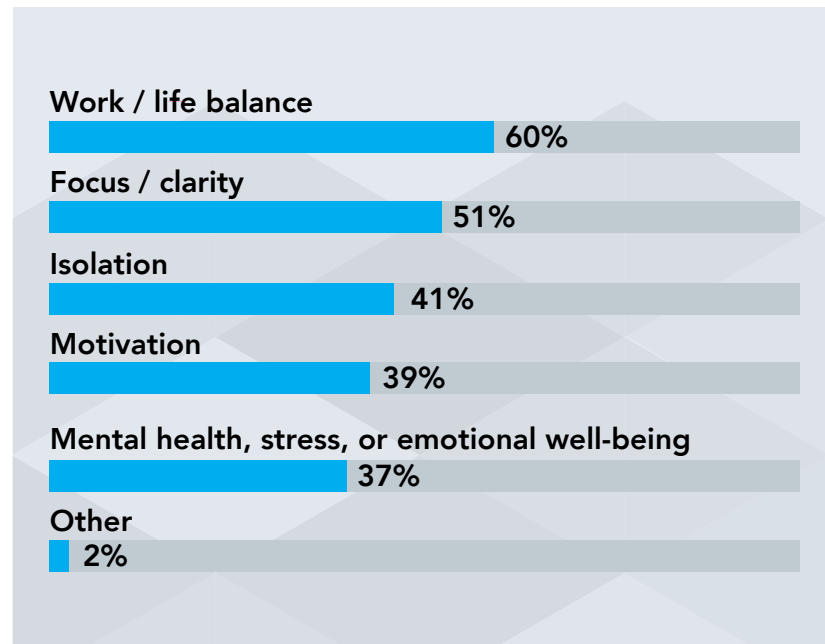
24%

### 3rd party ITIL v3 courses delivered in individual or group class settings



16%

In the 2019 SRE report, a large focus was on toil and stress. We quip with what may be some expected responses:



- ✓ Use automation to reduce toil.
- ✓ Be blameless to reduce post-incident stress.
- ✓ Shift to preventive measures through an observability capability to have less incidents in the first place.

## Which mental / personal well-being challenges are you facing since 'at home?

The forced working-from-home policy for most organizations has highlighted the need to pay more attention to the human well-being. As the reader looks at this at home dataset, ask, "How does a bullet from this dataset make better or worse a bullet from the beginning-of-the-year survey data?". For example, how would feeling isolated help or hurt someone who felt communications or lack of support were problems before? Would they likely feel more, or less, supported?

Turn newly-surfaced, or previously-ignored, challenges into opportunities for strategically differentiating. Focusing on challenges like morale, employee experience, work/life balance, and employee engagement & sentiment may showcase a company's employee-first mentality to attract or retain top talent.

## KEY TAKEAWAY 4

# The Future of SRE is Remote and Bright

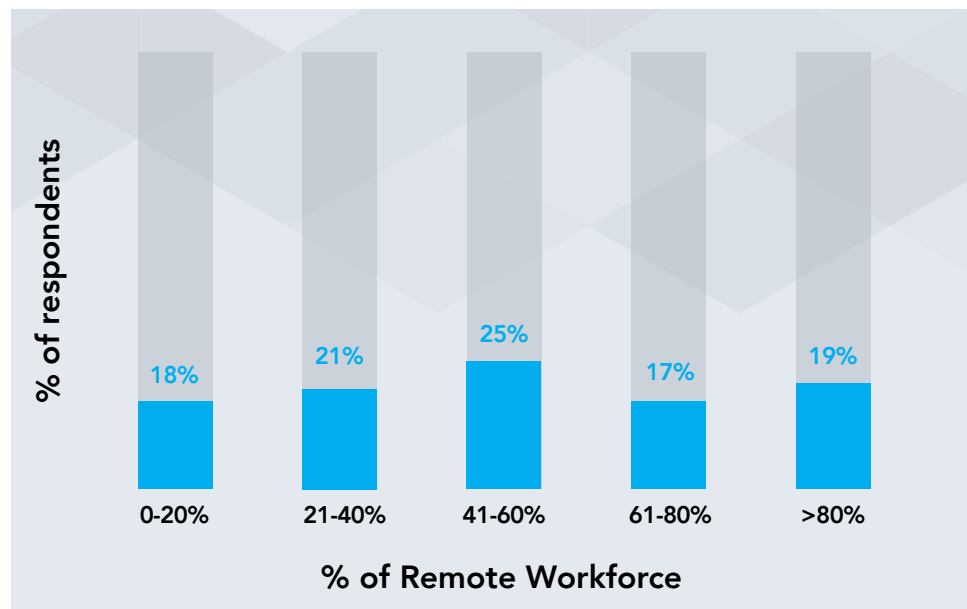
Re-evaluate various business continuity scenarios. Consider whether recovery times and recovery points need to be adjusted. When running your disaster or continuity exercises, identify areas of opportunity where preventive measures may now be implemented. Capture any new insights in your SRE charter.

IN OUR SRE 2018 SURVEY, THIS NEW YORK TIMES STYLE HEADLINE RAN:

**"If you're looking to work remotely the SRE role may not be the role for you. While some SREs work remotely, 81% of SREs state all or most of their team work in an office."**

The first question on our mind is, "When the world re-opens, what percent of your workforce will be "remote/work from home" first (What percent will be "onsite/in-office" second)?" In a time of asynchronous, worldly events, this data may not surprise you.

### Percent of expected remote workforce



Given the shift to a full, distributed workforce, we wanted to then look at other change factors to provide an input point for decision makers. What new challenges need to be addressed? What does 'managing incidents' look like? How would one run their disaster recovery table-top exercises if there is no table?

We don't want to make a "water is wet" type of statement when we say, "the future of SRE is remote". Rather, the future of SRE is remote and bright, with these caveats.

# "Grade school teachers aren't paid enough..."

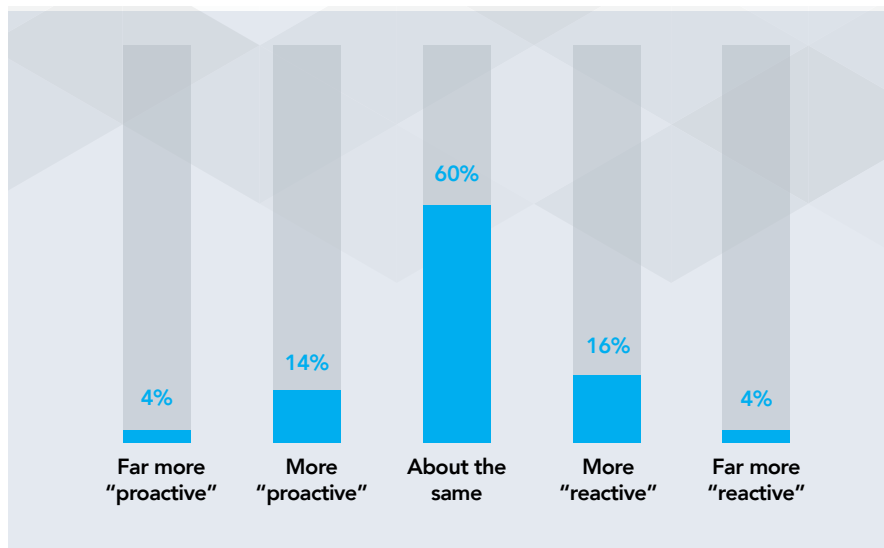
— SURVEY RESPONDENT



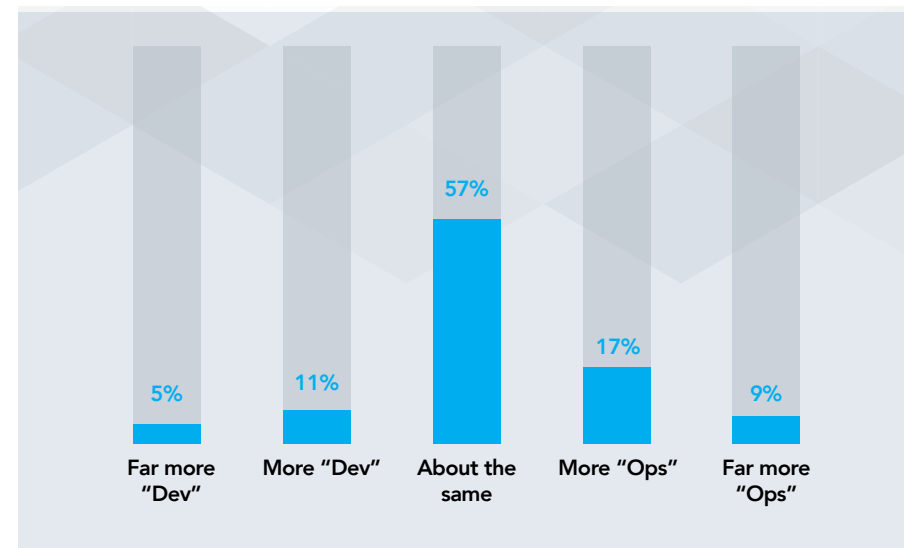
Keep in the mind this report's previous comments about the desire to prevent incidents by designing and building observable systems. Then consider how these direct, macro questions may affect your SRE charter.

The split between proactive versus reactive (net 2% toward more reactive) is not as large as the split between dev versus ops (net 10% toward more Ops). Here we again offer the pre-work from home data stating 75% of respondents were doing ops activities (versus 25% doing dev activities).

### How have your activities shifted since 'work from home'? (Proactive vs Reactive)

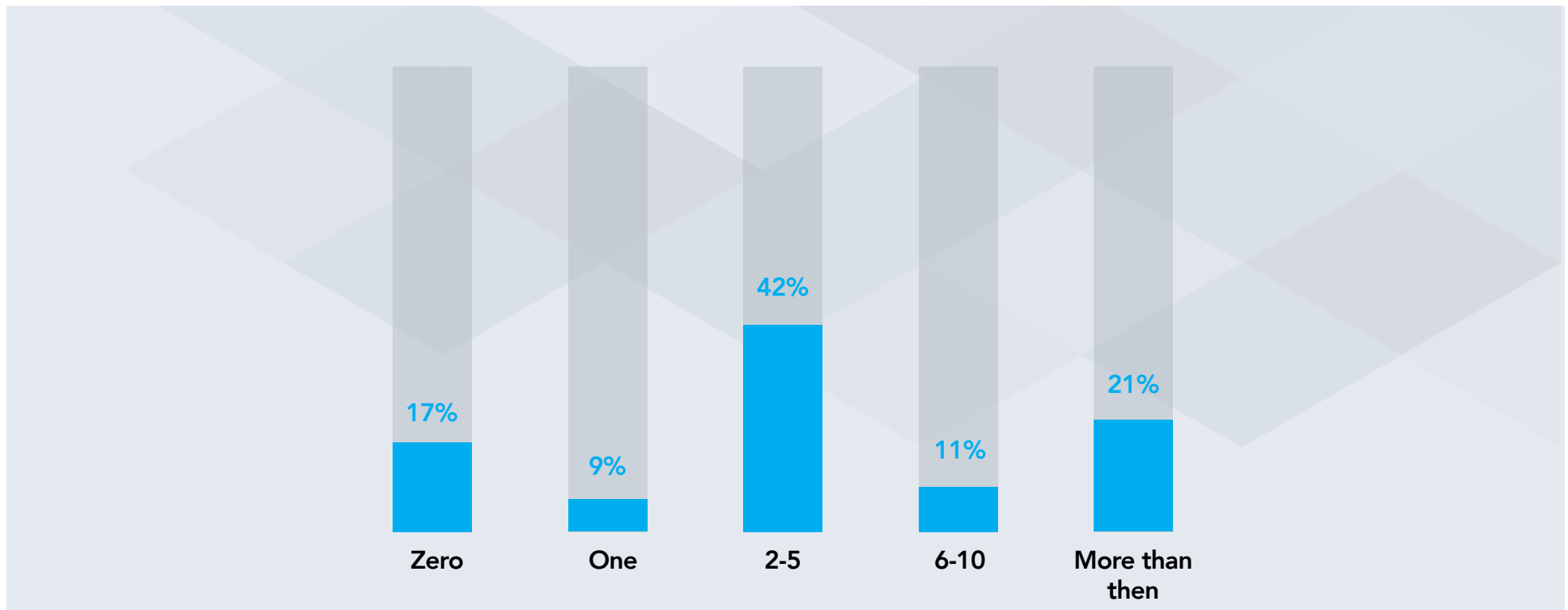


### How have your activities shifted since 'work from home'? (Dev vs Ops)



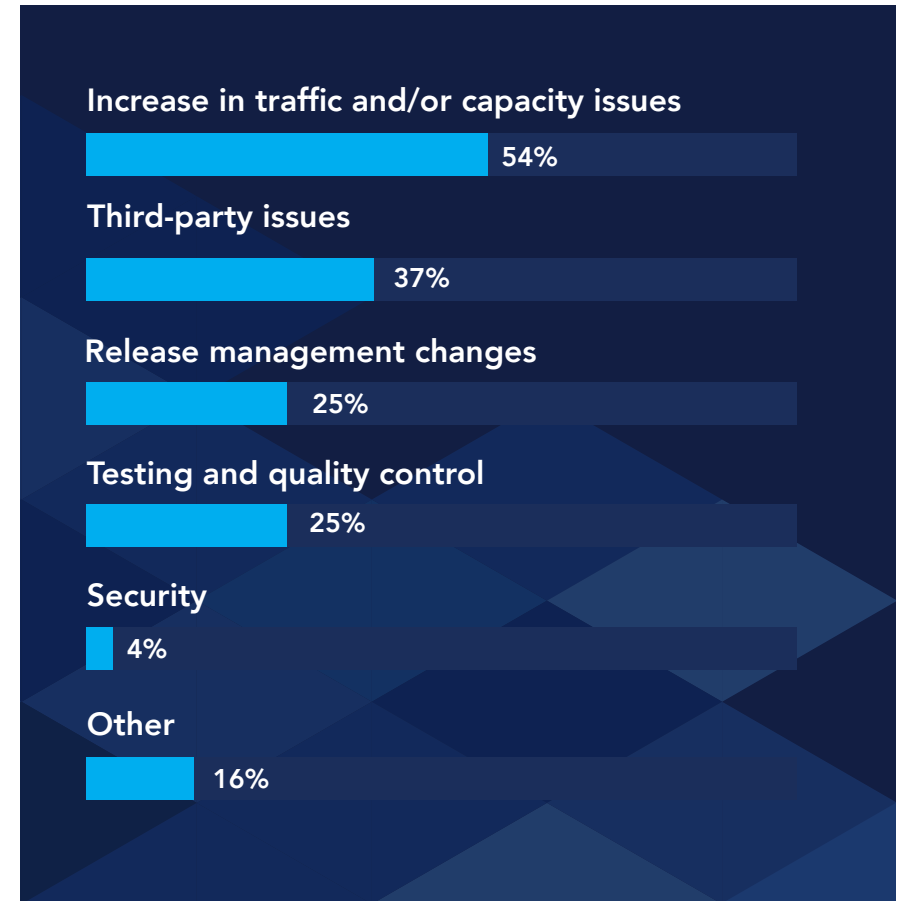
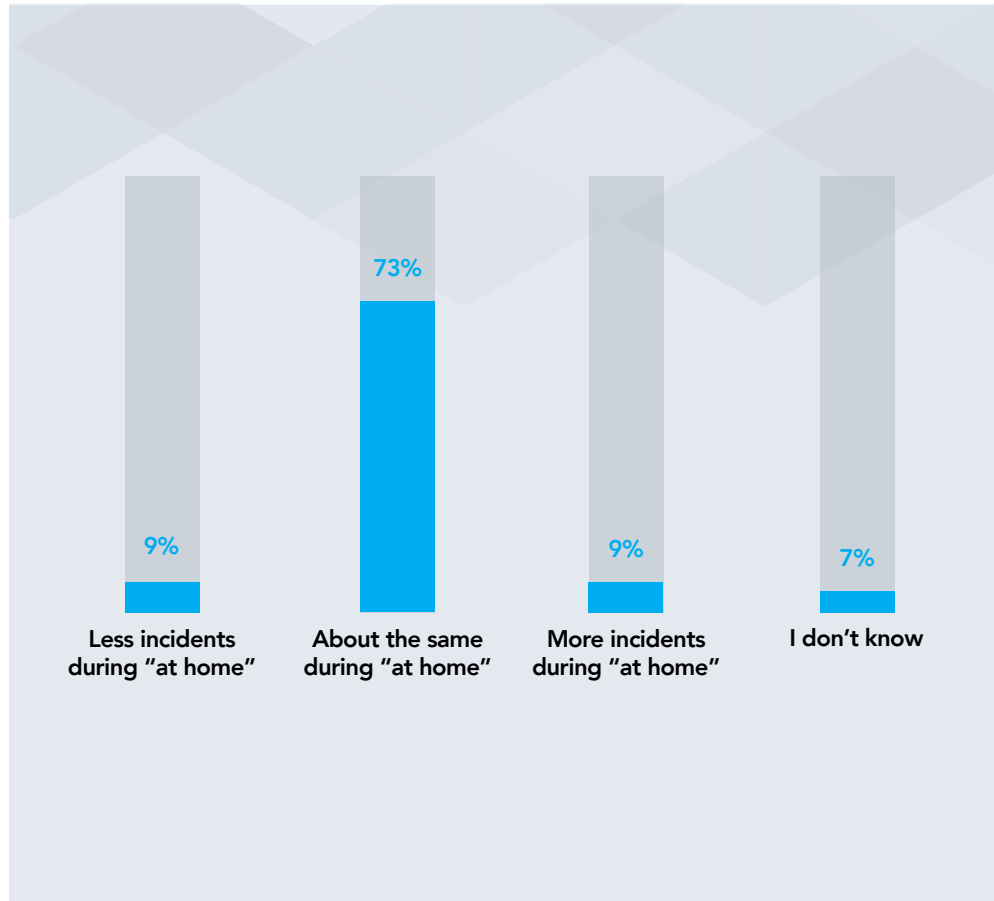
We wanted to get an idea of both the absolute number of incidents as well as the relative number of incidents (next page). As a reminder, the "post" set of survey questions was asked after two and a half months of being in the work from home state.

## How many incidents has your site experienced since work from home?



For the, "were there more or less incidents since at home" (next page), the data forms a normal distribution bell curve. What stands out from this question, though, is the 7% who do not know!

## Have your sites or apps experienced more or less incidents at home?



Increase in traffic and/or capacity issues was cited as the number one factor leading to increased incidents. Third parties were cited as the second most frequent factor, which is why we discussed the need to include a strategy for handling them in the first section of this report.

We would like to call out that only respondents who said they have had more incidents since work from home were asked this question. But we wanted to include the data for diligence.

A net +9% of respondents said managing incidents has become more effective since at home. This is a fascinating data and we wonder if there is a correlation of better incident management to companies doing less releases (according to this Atlassian data, 66% of respondents have slowed the frequency of their software releases). Note the 14% of respondents who chose unable to evaluate and identify opportunities to see if this is due to being at home.

## Please rate the effectiveness of your incident management process(es) since 'at home'.

Less effective while "at home" (higher MTTR)

5%

About the same MTTR

66%

More effective while "at home" (lower MTTR)

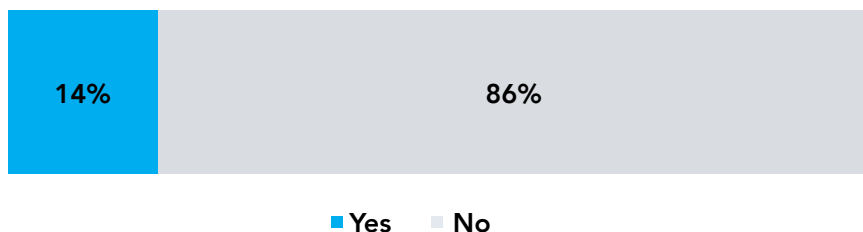
14%

Unable to evaluate

14%

The last direct question we asked in our 'at home' survey was, "Have you, or anyone on your team, had to be on site?" For the 14% who responded yes to this question, we then open-endedly asked, "How many times?". Their answers ranged from always and one person per shift (follow the sun) to just once and every couple of weeks.

### Have you had to be on site since 'at home'?



### What aspects of incident management has become more challenging since 'at home'?

Detecting incident occurrence/outages



Identifying incident root cause



Escalating to the right teams



Fixing the root cause



Validating whether fixes were successful



None of the above



As we work to close this year's report, we offer one last datapoint while companies re-evaluate how they will continue business. We asked, "How often are you conducting disaster recovery scenarios since at home?" As you consider how your various recovery times may be affected in the event of a disruption, consider the previous path to preventive motto as you work to design and implement observable systems.

Observability is all about being able to answer, "Why is our customer's experience the way it is?" Is it because of a third party, application code, transit network, or other delivery chain component like DNS or CDN? Then use those answers to iterate on improving either existing or new products or services.

When 'building in' reliability, consider the split between development versus operational work as you develop your SRE charter. The goal here is to include reliability as early as possible as it is much easier to improve reliability of an already reliable system.

Last, consider the distributed nature of your workforce and acknowledge sets of challenges which may have been previously-ignored or non-existent. Things like toil, lack of support, work/life balance, and feelings of isolation may cause certain playbooks or processes to be re-evaluated from the ground up.

## How often are you conducting disaster recovery scenarios since 'at home'?

Never



Not yet; we are planning



Randomly



Monthly



Weekly



Other





## Methodology

In January 2020, Catchpoint conducted an SRE survey promoted via email lists and social media. The survey questioned technical professionals from across a variety of industries about their role as a site reliability engineer. Through the report, this set of questions is referred to as the "pre" set of questions.

In June 2020, Catchpoint conducted an addendum survey to include consideration for worldly events regarding the COVID 19 stay-at-home mandate. This set of questions was designed to ask various "what has changed" questions and is referred to as the "post" or "at home" set of questions.

At the time of this report writing, there were a total of 594 survey respondents. Additional responses trickled in the time between formatting the report and authoring the appendix, but they only affected the statistics in this report by less than 1%.