



2020 Report:

The State of Software Quality

OverOps

Table of Contents

- 3 | **Executive Summary & Key Findings**
- 4 | **Methodology & Demographics**
- 5 | **Speed vs. Quality Showdown:**
Engineering Teams Are Driven by Quality
but Distracted by Speed
- 11 | **The Modern Software Quality Ecosystem:**
Top Techniques, Tools & Metrics Used Today
- 16 | **2020 Roadmap:**
Automated Code Analysis Emerges as the
Next Big Thing in Software Quality
- 19 | **Conclusion**

About the Report

Teams tasked with developing and delivering software are under pressure to balance the business imperative for speed with high customer expectations for quality. In the course of trying to achieve this balance, engineering organizations rely on a variety of tools, techniques and processes. The State of Software Quality report provides a snapshot of the key challenges organizations encounter when it comes to delivering quality software at speed, as well as how they are approaching these hurdles.

The data in this report is based on a survey of hundreds of software professionals across all major industries, including financial services, technology, retail and telecommunications. The majority of participants are individual contributors or managers in engineering, ITOps and QA roles at large enterprise companies based in North America and Europe.

Executive Summary

In today's software delivery pipeline, speed and stability are frequently at odds. The fast-paced business landscape has driven the need for increased development velocity – but often at the expense of application quality.

Engineering teams are thus challenged with the impossible task of keeping up with competitors without sacrificing the customer experience in the process. Faced with this dilemma, some organizations opt for prioritizing speed over quality, while others double down on testing and observability in the hope of preventing critical issues in their live environment.

**But which approach is the right one?
Or is it possible to strike a balance
between the two?**

Agile adoption is alive and well across all industries and company sizes. According to the survey, the majority of respondents – regardless of company size, DevOps adoption, industry or infrastructure – are running more frequent release schedules. More than 90% are deploying code at least once a month, and over 60% are deploying code at least once every two weeks. 43.8% of all respondents also noted that they align their code or feature releases with sprints.



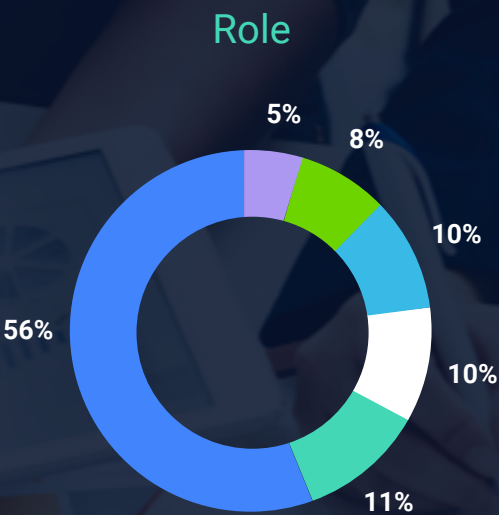
Key Findings

- 01. Speed vs. Quality Showdown: Engineering Teams Are Driven by Quality but Distracted by Speed**
 The overwhelming majority of engineering organizations (70%) say software quality trumps delivery speed, yet critical production issues persist. The survey found that despite most teams making quality a priority, a lack of processes, pressure to move fast and other factors hurt developer productivity and lead to over 50% of teams experiencing customer-impacting issues once or more per month.
- 02. The Modern Software Quality Ecosystem: Top Techniques, Tools & Metrics**
 From CI/CD and daily release schedules to feature flags and alerting, today's engineering teams are using a broad ecosystem of cutting-edge technologies and practices. According to respondents, some of the top tools include manual testing (70%), automated testing (65%) and log management (62%). But failure to invest enough code analysis and observability could be causing software quality efforts to still fall short.
- 03. 2020 Roadmap: Automated Code Analysis Emerges as the Next Big Thing in Software Quality**
 Is this the decade automation will transform the way we see software quality? Survey results show that while engineering teams are continuing to invest in pipeline automation and containerized microservices, automated code analysis sees an uptick as an emerging technology. When asked about which tools participants plan to adopt in 2020, 37% said static code analysis and 28% said dynamic code analysis, putting these analysis tools at the top of the list.

Methodology & Demographics

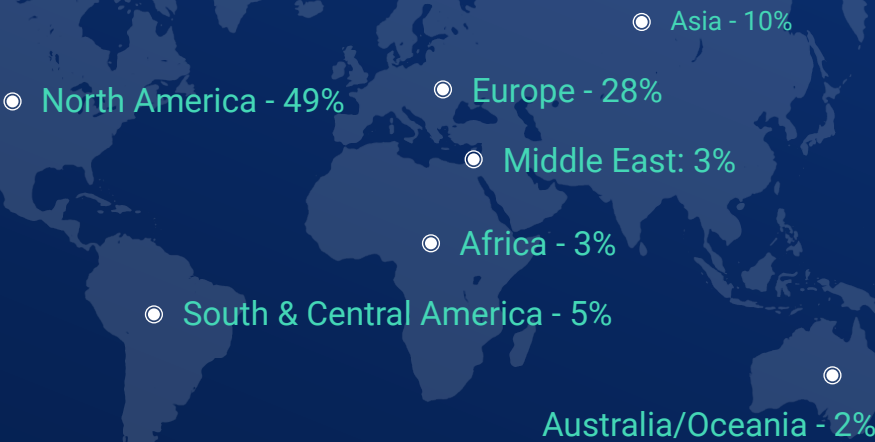
This report is based on a survey conducted by OverOps of over 600 IT professionals ranging from developers and QA professionals to DevOps engineers and SREs. We gathered responses through a variety of channels, including including data bases of software developers and leaders, social media and third-party websites geared toward engineering professionals.

Current State of DevOps Adoption



- Developer/Software Engineer
- IT Executive (CIO, CTO, VP, Director)
- DevOps
- IT Ops/Site Reliability Engineering
- QA
- Other

Region



Top Industries

- Technology
- Finance/Financial Services
- Public Sector
- Retail/eCommerce
- Telecom



01. Key Finding

Speed vs. Quality Showdown: Engineering Teams Are Driven by Quality but Distracted by Speed

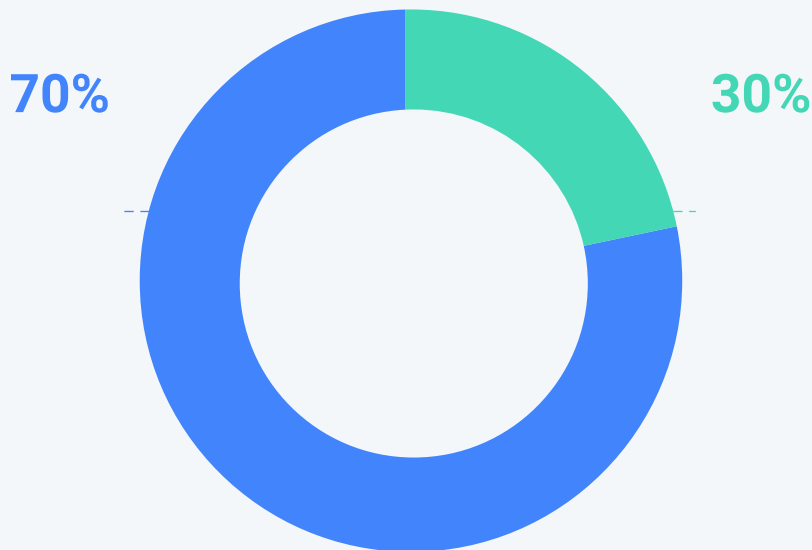
We asked survey participants to share how they approach the speed vs. stability paradox, as well as some of the challenges they encounter when trying to deliver reliable software. While the overwhelming majority of organizations prioritize quality over speed, they continue to encounter challenges that result in frequent critical issues in production and significant customer impact.

▶ Suggested Viewing

You've heard of CI, you've heard of CD – but what about CR? Watch this webinar to learn more about Continuous Reliability and how it can help your team address the speed-stability paradox. *The DevOps Paradox: How to Balance Speed and Quality with Continuous Reliability*

The Speed vs. Quality Showdown

When asked which statement most closely aligns with their team's approach to software quality, **the majority of survey participants (70%) indicated that quality is paramount**, meaning they cannot afford production issues and would rather delay the product roadmap than risk a critical error impacting their users. Only **30% said they would rather get releases out the door as fast as possible**, even if that means critical errors sneak by into production.

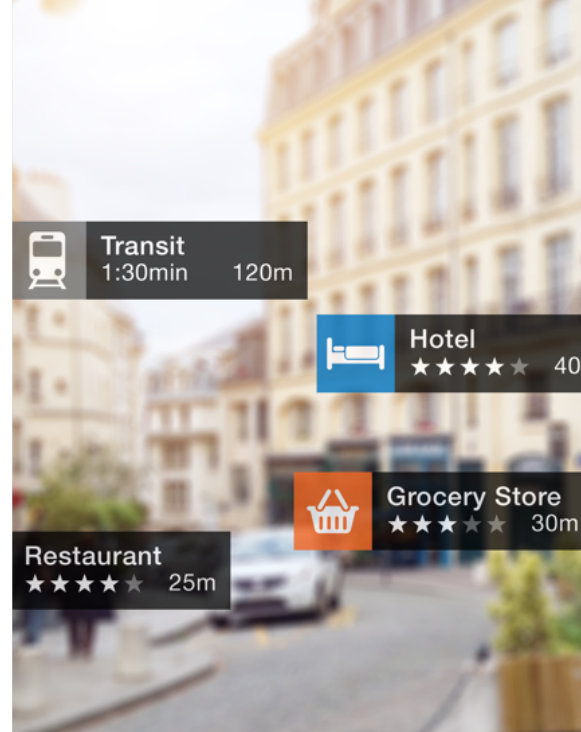


Quality is Paramount

We cannot afford production issues, so we test as much as possible in pre-production, even if that means slight product/feature delays

Speed is Paramount

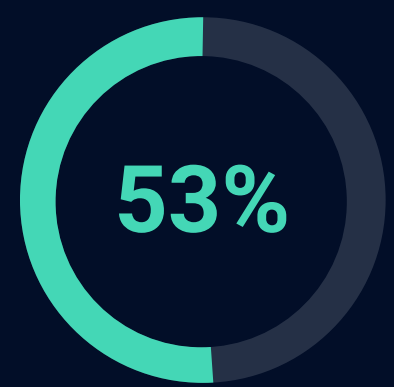
We like to get releases out the door as fast as possible, even if that means a few critical errors sneak by sometimes



Delivering High-Quality Applications is a Challenge

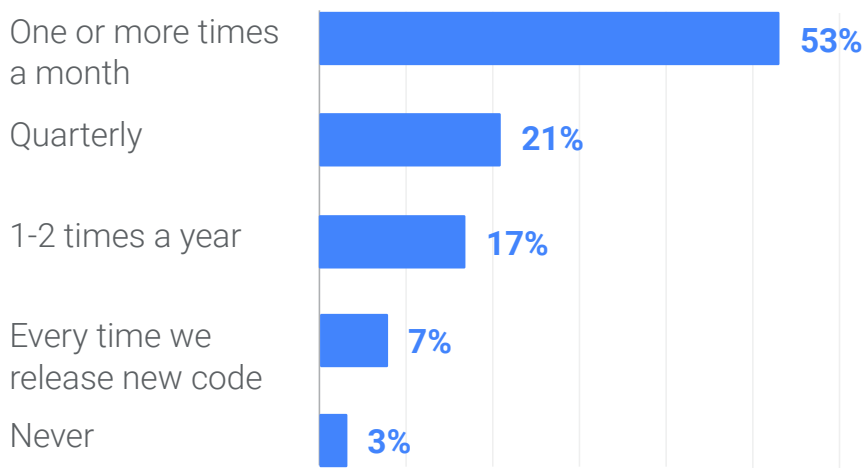
Regardless of whether respondents care more about speed or about quality, across the board they are encountering challenges in delivering reliable software. **Over half of survey respondents (53%) indicated that they encounter critical or customer-impacting issues in production at least one or more times a month.**

Compounding the impact of these critical production errors is the frequency at which they are first detected by customers, rather than by internal error detection mechanisms. **Nearly a quarter of survey respondents said that over 40% of critical production issues are first reported by end users or customers** rather than by internal tools or processes.

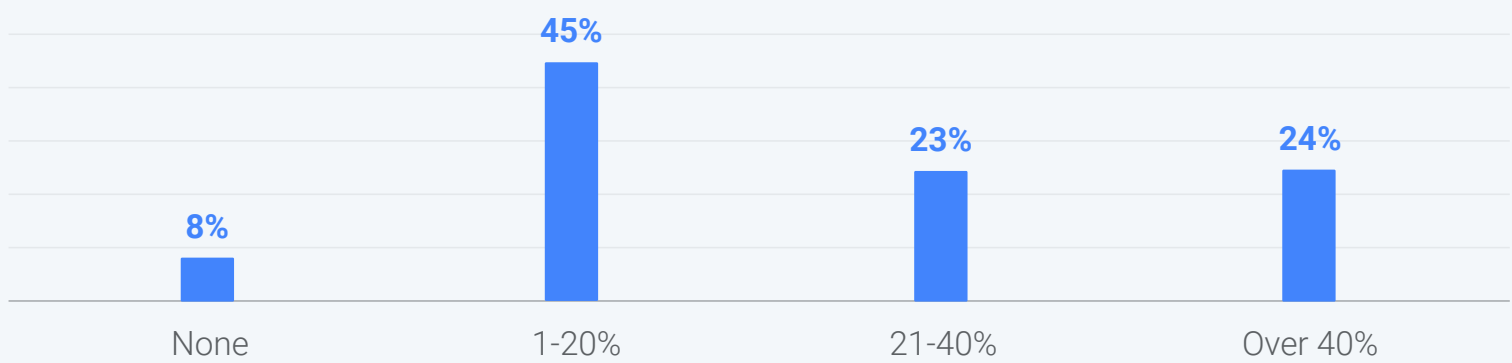


Over half of survey respondents indicated that they encounter critical or customer-impacting issues in production at least one or more times a month.

Frequency of Critical Production Errors



Approximately what percentage of customer-impacting incidents are first reported by end users (i.e. before tooling or internal processes can catch the issue)?



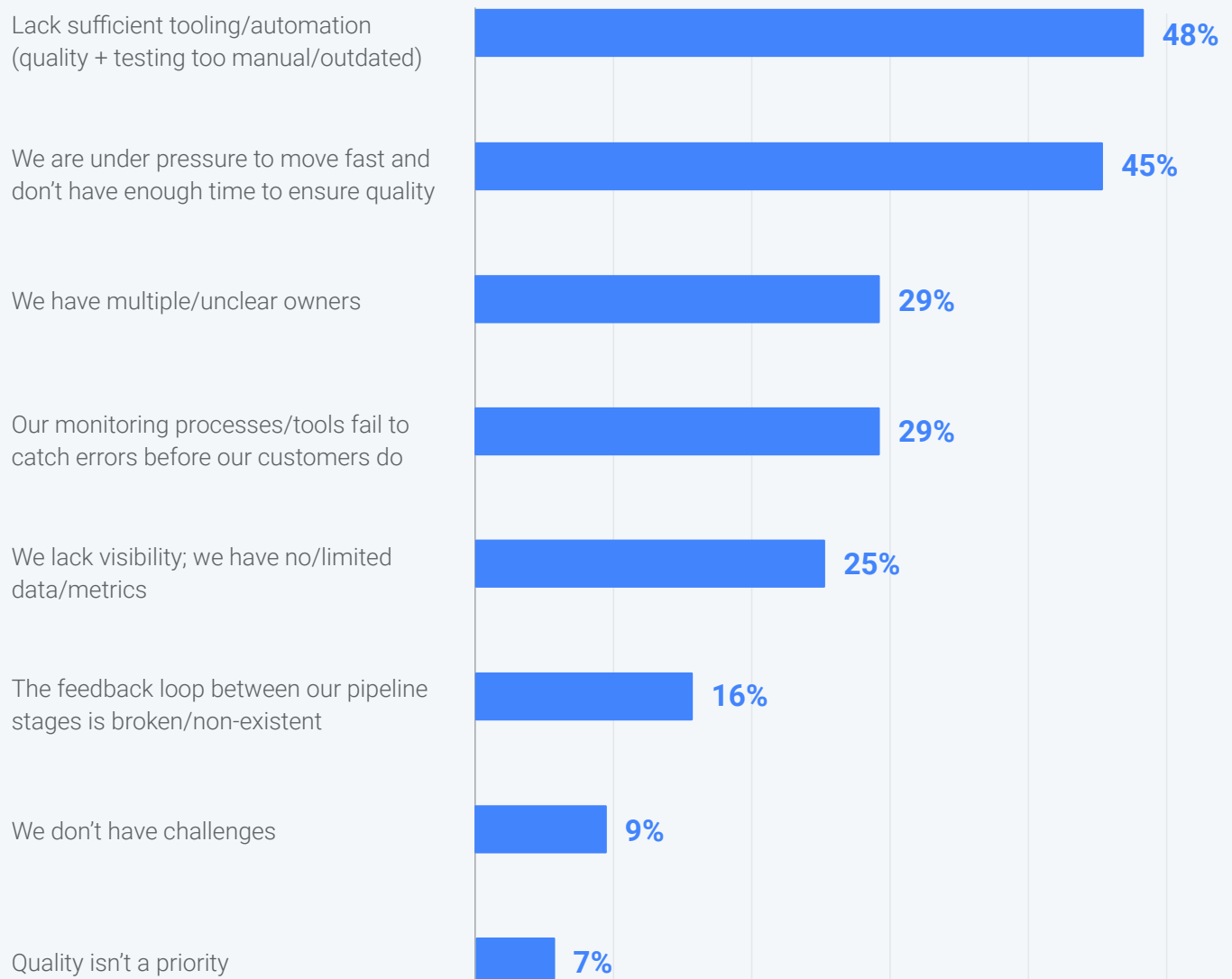
Top Application Quality Challenges

When asked about their top application quality challenges, **nearly half of participants (48%) cited manual processes and a lack of sufficient tooling as a major hurdle.** This aligns with some of the other top challenges identified by participants, including monitoring processes and tools that fail to catch errors before customers (29%), and a lack of visibility, data and metrics (25%).

Despite only 30% of respondents indicating they prioritize speed, **45% of all respondents said they were under pressure to move fast,** leaving them with insufficient time to ensure quality.



Top Application Quality Challenges



Process & Culture are Key to Eliminating Errors

To get to the root of why production errors continue to occur, we asked participants about the underlying causes of software quality issues. **Less than a quarter of respondents (20%) indicated that these issues happen by design** – i.e. they have adopted the “test in production” philosophy – while the other 80% of respondents cited several reasons that errors unintentionally make it to production.

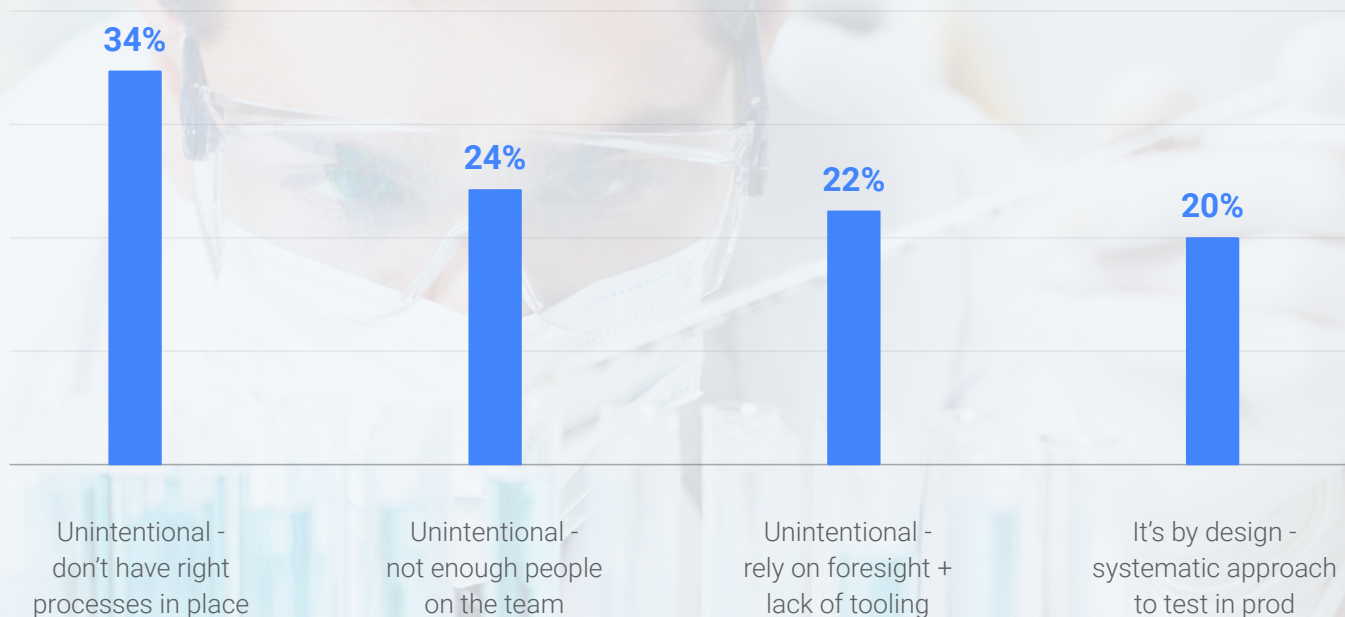
Leading the pack was a **lack of proper processes and culture (34%)**. Further, respondents who stated they believe that speed is paramount were 70% more likely to cite lack of process as the primary reason for errors.



📁 Pro Tip

The Rule of Ten in software says that the cost of finding and fixing defects increases 10x the further into the software delivery lifecycle you get. By adopting a shift left approach to quality, you can catch critical application errors before they reach production and limit the cost to your organization.

What is the primary reason errors make it into production at your organization?



Developer Productivity is Suffering

As a result of frequent critical production errors, development teams are spending a considerable amount of time troubleshooting code-related issues.

Two out of three survey respondents report spending at least a day per week troubleshooting issues in their code, with 30% spending anywhere from 2 days to a full week.

📖 Suggested Reading

Intuit's engineers significantly improved their troubleshooting workflow, allowing them to innovate quickly without compromising on the quality of their products.

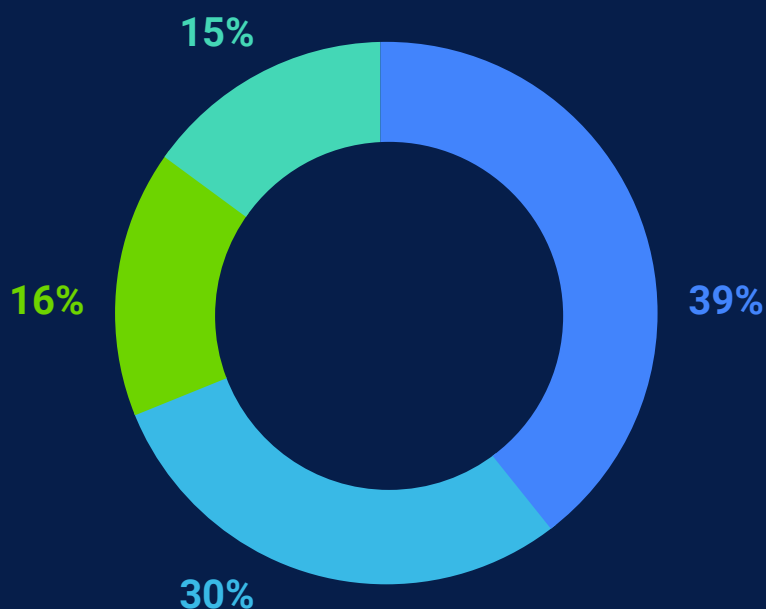
[How Intuit Engineers Manage to Meet Product Deadlines with Time to Spare](#)



39%

of respondents spend one day per week troubleshooting code-related issues

How much time do you spend troubleshooting code-related issues?



1 day per week

It's not my responsibility

2 or more days per week

Less than a day per week (we barely have errors)

```
T> PaginatedScanList<T>
DynamoDBScanExpression
if (condition == null)
    scanExpression = n
}
else {
    scanExpression = c
}
if (limit != null) {
    scanExpression.set
}
final DynamoDBScanExpr
return this.executeWit
return this.mapper
```




02. Key Finding

The Modern Software Quality Ecosystem: Top Techniques, Tools & Metrics

How are organizations addressing quality today? Where are the gaps? We dug into the common tooling and processes used by today's engineering teams and found that some of their software quality investments contradict their primary goals. Even with broad ecosystems of tools and cutting-edge procedures, their efforts still fall short in ensuring software quality.

📖 Suggested Reading

As your team grapples with the challenge of balancing speed and quality, it may be time to evolve some of your practices and tooling. Check out this report about DevQualOps, a new way of thinking about testing and quality in the context of your DevOps pipeline.

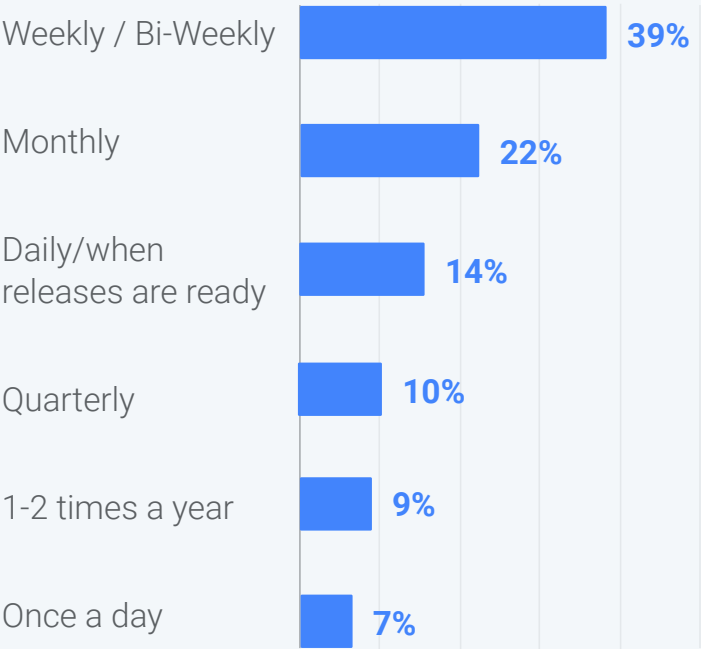
Delivering Software at Speed Without Sacrificing Quality

Engineering Teams Fail to Put Their Money Where Their Mouths Are

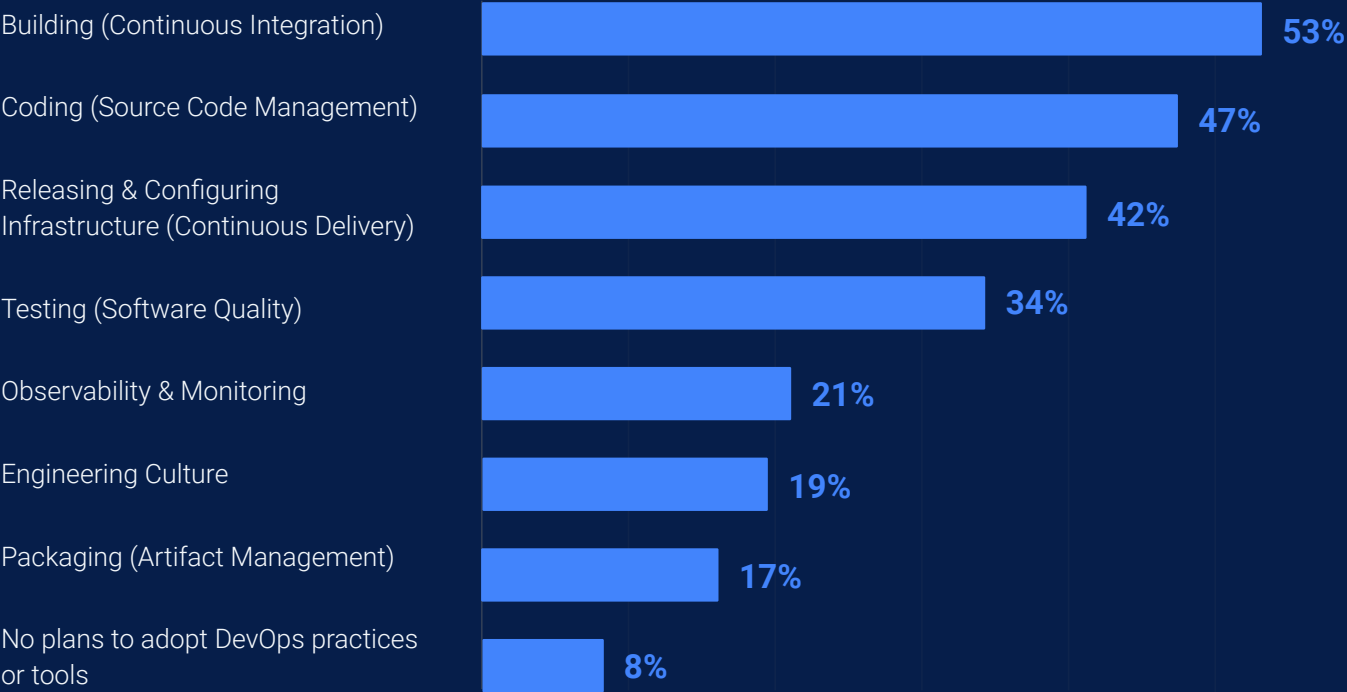
Despite survey respondents reporting that quality is more important than speed, their key areas of DevOps investment indicate otherwise. Continuous Integration and Continuous Delivery, both hallmarks of accelerated software delivery pipelines, were among the top areas of DevOps investment, at 54% and 42%, respectively. Meanwhile, **testing/software quality and observability and monitoring, key components of software quality success, fell lower on the list, at 34% and 21%, respectively.**

In support of this finding, the overwhelming majority of survey participants also indicated they are releasing software at a high velocity. **Over half of respondents (59%) said they release new code/features anywhere from bi-weekly to multiple times a day**, while only 19% follow a quarterly or less frequent cadence.

Release Frequency



Top Areas of DevOps Investment



Pre-Production Pipeline: Testing and Static Analysis are Top Software Quality Techniques

We asked survey respondents to share the various tools and techniques they use to ensure application quality across the software delivery life cycle. **In pre-production, the overwhelming majority of respondents rely on a mix of manual and automated testing.** Other top tools and methods included manual and automated code analysis, log management, alerting and load testing.

📖 Suggested Reading

Read this blog post about how the move to innovate at speed and scale is stressing software quality and exposing the limitations of traditional testing pipelines.

[Why Testing is No Longer Sufficient for Today's Software Delivery Pipelines](#)

📁 Pro Tip

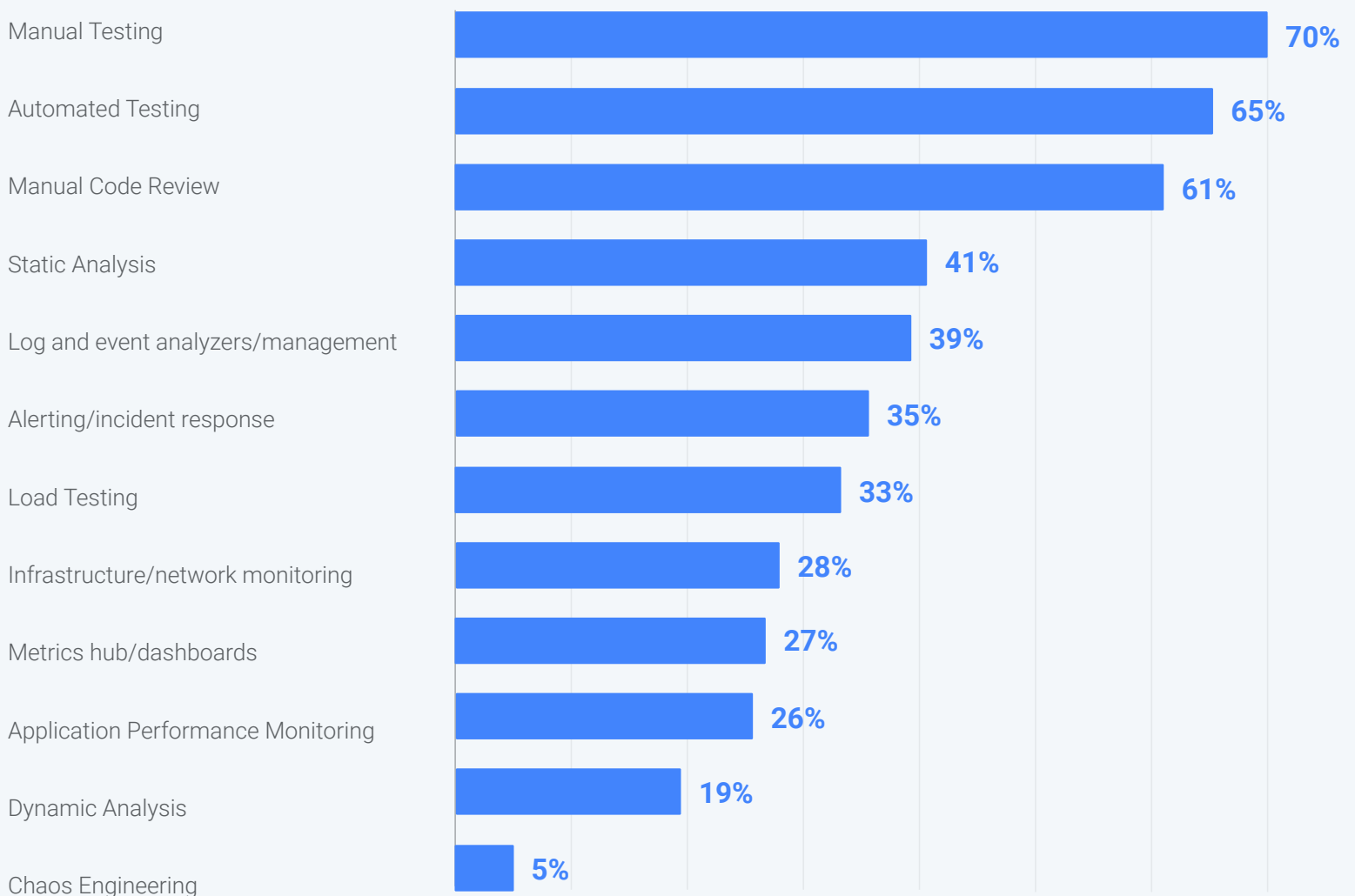
Log files help identify when or where something has gone wrong, but they typically don't tell you why or how. Tools like dynamic analysis can help fill in the the gaps left by logs to ensure developers have complete context about errors and their cause.

Manual testing is the

#1

tool used in pre-production
to ensure quality

Pre-Production Quality Tools & Processes



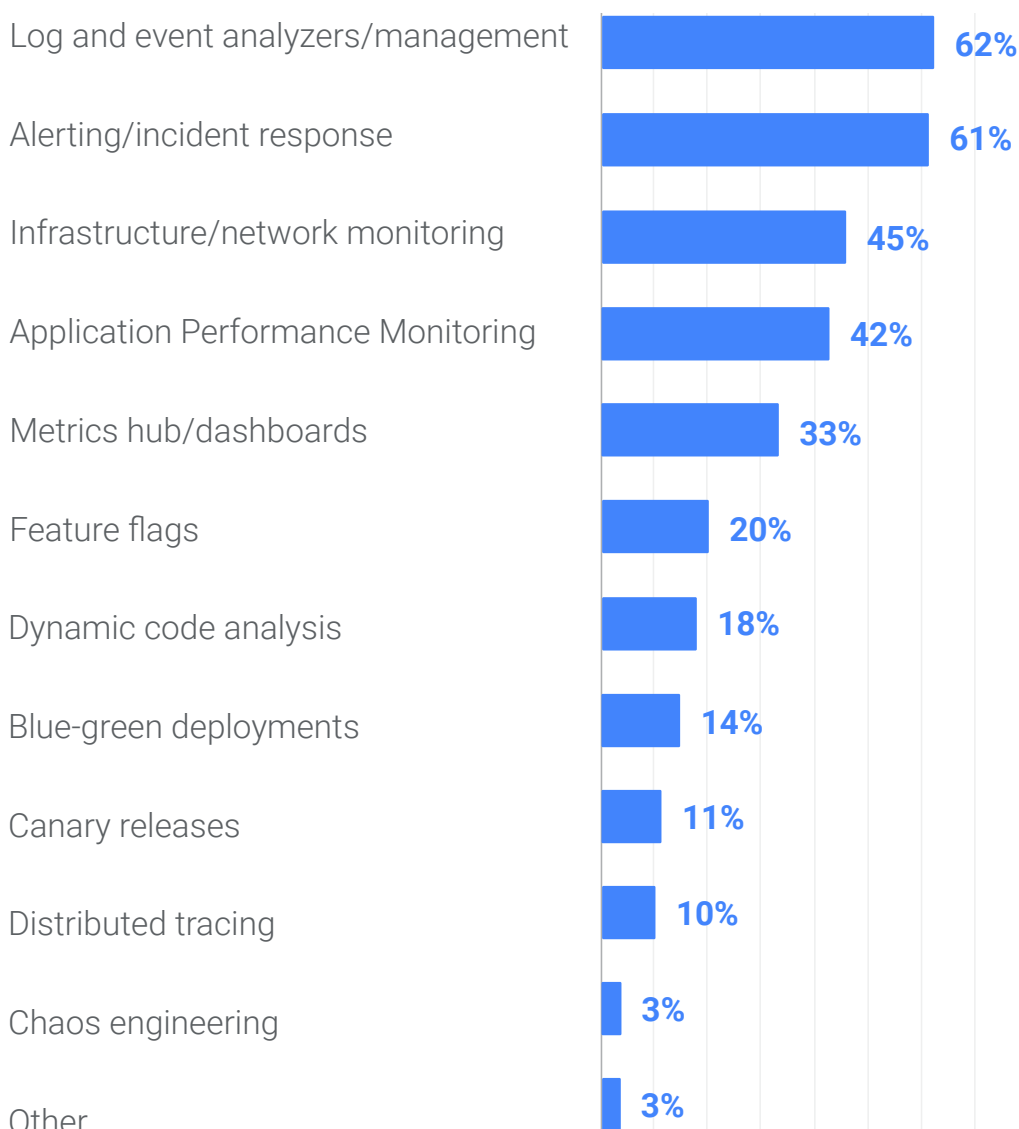
Production Pipeline: Logs and Alerting Reign Supreme

On the production side of the pipeline, **log management and alerting tools reign supreme**, with 62% and 61% of participants **respectively relying on these tools**. Other critical production quality tools include infrastructure and application performance monitoring and metrics hubs.

62%

of respondents use log management tools to ensure quality in production

Production Quality Tools & Processes



📖 Suggested Reading

Despite their pervasiveness, log files and their associated tools often lack the depth and context needed to effectively determine the root cause of an issue.

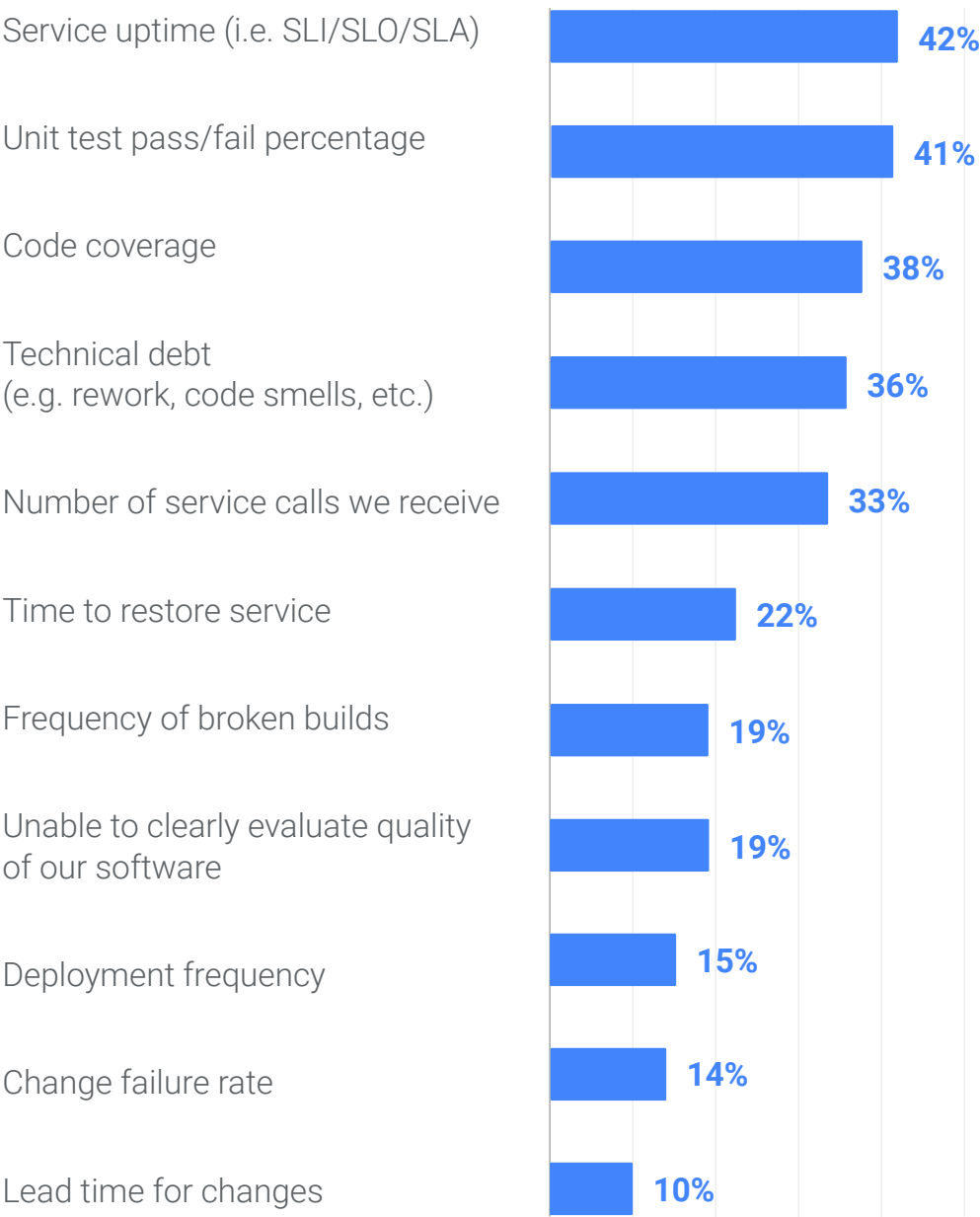
[The 7 Key Components of True Root Cause Analysis](#)

There's No Universal Way to Assess Software Quality

When asked which indicators/metrics survey participants use to evaluate the quality of their applications, **service uptime (42%)** and **unit test pass/fail percentage (41%)** took the top spots, although still leveraged by less than half of all respondents.

Other top methods of quality evaluation included code coverage (38%), technical debt (36%) and the number of service calls received (33%).

Application Quality Indicators & Metrics



Suggested Reading

If your team is struggling with measurement, this Continuous Reliability Maturity Model can provide a North Star for where to invest resources and how to evaluate success.

The Complete Guide to Delivering Reliable Software in the Enterprise



03. Key Finding

2020 Roadmap: Automated Code Analysis Emerges as the Next Big Thing in Software Quality

Engineering organizations are expanding the scope of their automation initiatives beyond their CI/CD pipelines and containerized microservice infrastructures, going deeper into the realm of code quality. While DevOps and microservices remain the top priority for engineering organizations in 2020, automated code analysis is emerging as the next technology that will drive bottom line growth.

📖 Suggested Reading

Read this blog post to get to know the different types of automated code analysis when to use them.
Static vs. Dynamic Analysis: Two Foolproof Ways to Ensure Your Code is Production-Ready

Quality is a Top Priority for Engineering Organizations in 2020

Most software engineers and executives seem to agree that code quality and productivity are important. Over half of all respondents indicated that improving code quality (60%) and increasing developer productivity (52%) are their company's primary software quality goals for 2020. **Only 6% of respondents replied that quality is not a priority for their engineering organization in 2020**, indicating that the vast majority of organizations are making software quality a priority to some degree.

Reducing technical debt comes in at #3 with 40% of respondents indicating that it's one of their primary goals. Loosely defined as the estimated amount of effort for reworking existing solutions in a more efficient way, its influence on quality is tied to the long tail of issues companies are experiencing.

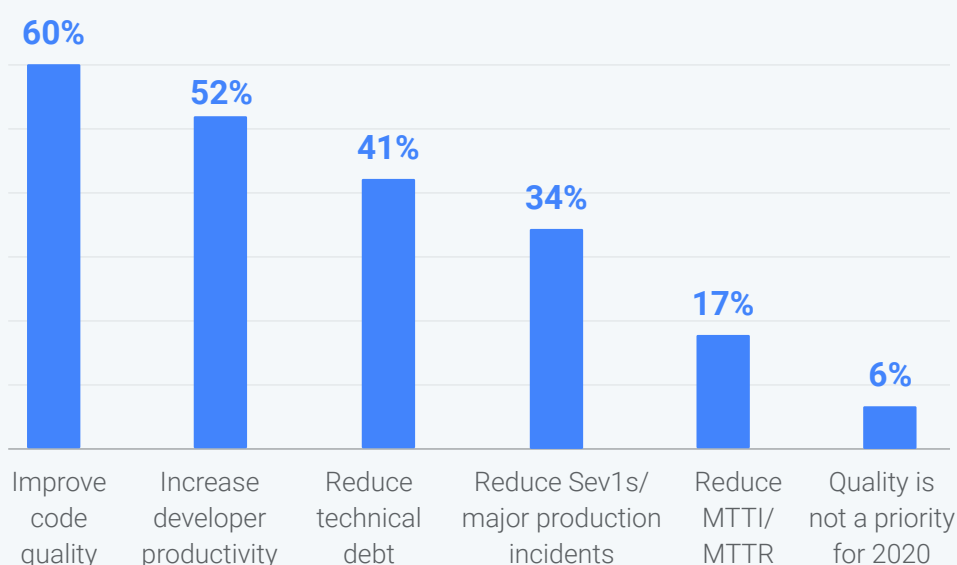
Reducing production incidents (34%) and lowering the Mean-Time-To-Identify and Resolve issues (17%), traditionally operations related goals, are closing the list and showing that while quality is thought of as a developer concern, it has significant implications on the user experience in production.



94%

of organizations are investing in software quality initiatives in 2020

What are your engineering organization's primary goals for 2020 as it relates to software quality?



📁 Pro Tip

Setting up quality gates for new errors between pipeline stages adds a layer of insight for go/no-go decisions. And the underlying technology for quality gates is often based on testing, static and dynamic analysis.

Automated Code Analysis Emerges as the Next Big Thing

Automated code analysis is becoming the next area of interest for engineering organizations in 2020. **Both static and dynamic analysis techniques are increasing in popularity**, helping engineering organizations reduce their reliance on manual processes that are prone to human error.

DevOps (57%) and Microservices (45%) continue to grab most of the mindshare in the engineering community. Fueled by an increase in open source cloud native technologies like **Kubernetes** and **service mesh** solutions.

On the long tail of the techniques mentioned by the survey's respondents, SRE (20%) and Observability (17%) came in as the 3rd group of practices that were mentioned. **Despite industry hype, only 12% or less indicated plans to adopt shift left testing, shift right testing, chaos engineering and AIOps.**

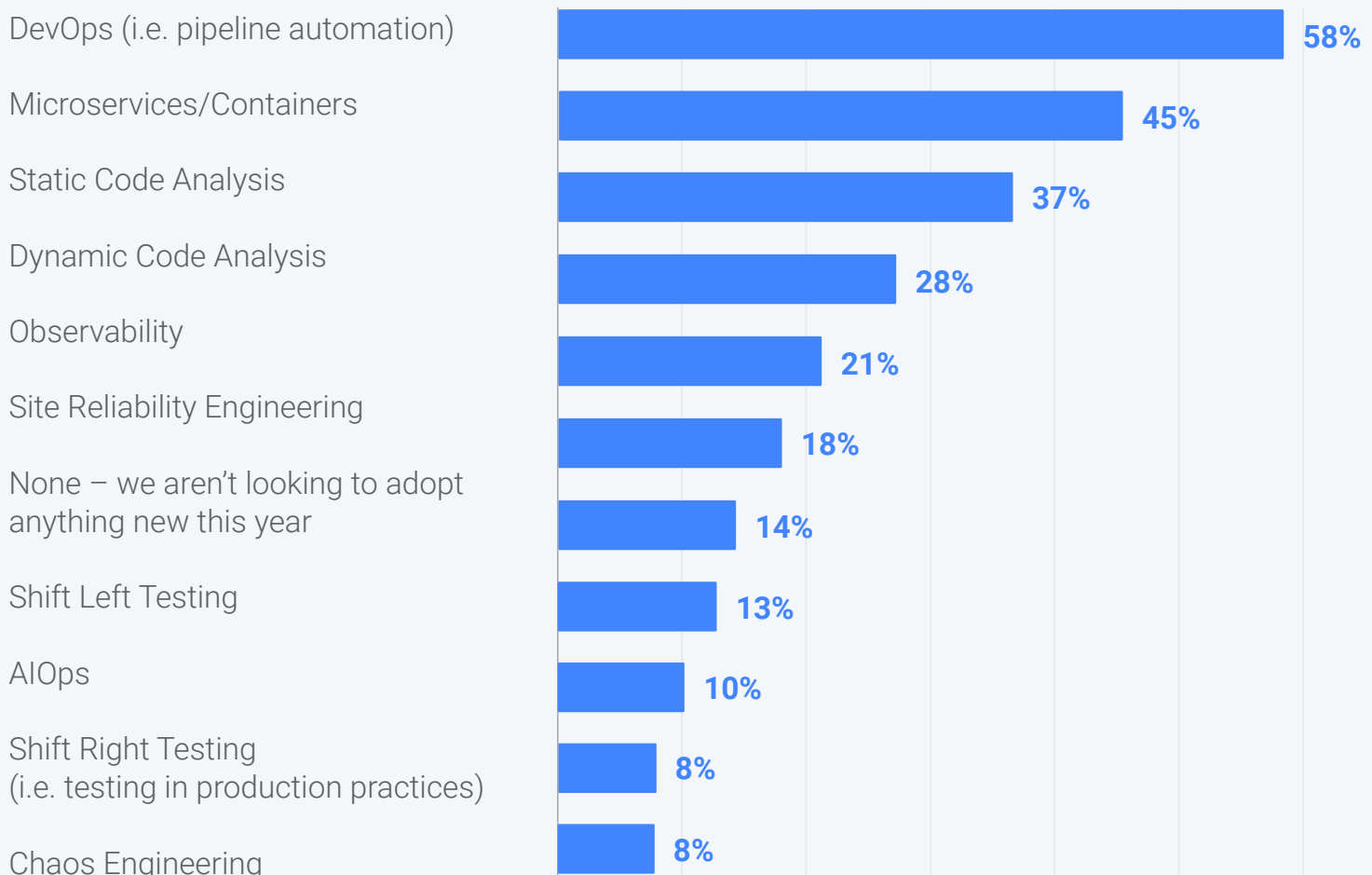
▶ Suggested Viewing

*Static vs. Dynamic Analysis:
Two Foolproof Ways to Ensure Your
Code is Production-Ready*

🎁 Pro Tip

When evaluating new solutions try to focus on how they fit within your desired workflow and what differentiates them from the way you're working today.

2020 Tool and Technique Adoption



Conclusion

Today's Software Development and Delivery Teams Have a Long Way to Go When It Comes to Striking a Balance Between Speed and Quality.

Regardless of priorities, very few organizations have been able to find a balance that allows them to deliver software at speed without encountering critical errors in production on a regular basis. At the root of this problem seems to be a misalignment between processes and resources with primary engineering and business goals. This is compounded by failure to invest in the right tooling and culture that enables efficient error detection and resolution, regardless of where you are in the pipeline.

Recommendations:

- For organizations that prioritize speed over quality, having a sound shift right strategy will ensure that you have a staging and production mechanism for detecting and addressing issues to counteract any mishaps that occur as a result of fast release cycles.
- For organizations that absolutely cannot afford production errors, the more automation you can incorporate into your pre-production pipeline, the faster you will be able to test quality with a high degree of confidence. These teams should focus on shifting left and invest in a network of software quality tools like static and dynamic analysis that integrate seamlessly into a CI workflow.
- As a first step, we advise experimenting with new technologies and increased automation in lower environments such as testing or staging. Once your team is comfortable with the new solution and can demonstrate its value clearly, gradually replicating the success to additional environments is the next logical step.



About OverOps

OverOps is a continuous reliability solution that enables companies to ensure rapid code changes do not impact customer experience. Using OverOps, teams can quickly identify, prevent and resolve critical software issues. Unlike static code, log analyzers and APMs that rely on foresight, OverOps analyzes your code at runtime to deliver deep insights into when, where and why code breaks. With robust CI/CD integrations and open APIs, OverOps ensures software reliability from testing to production. The company is backed by Lightspeed Venture Partners and Menlo Ventures, and has offices in San Francisco, Orlando and Tel Aviv.

www.overops.com

