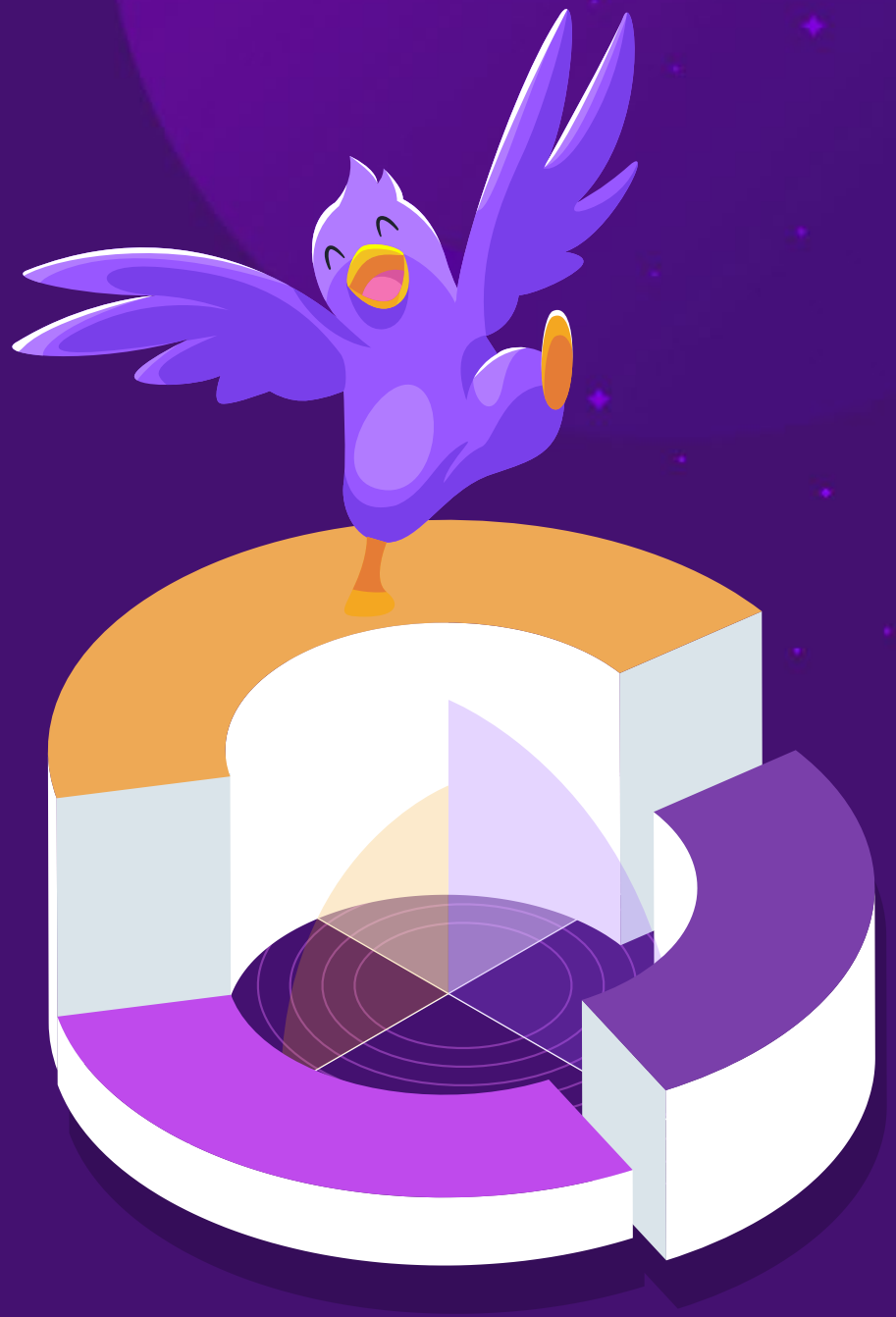
ROOKOUT

REPORT

# Top Digital Transformation Challenges for Engineering Managers



# Table of Contents



# Introduction

A digital transformation towards distributed technologies and workflows has been occurring in the past few years. These advances have caused a fundamental change in the way that developer teams work, enabling the building of bigger, more complex systems. Yet, with the good always comes the difficult, and so too is the case with this digital transformation. Software engineers have needed to become more productive and agile, but while doing so, are faced with the unique challenges that distributed environments and distributed teams pose. We surveyed the leading managers to find out exactly what those challenges are and what can be done to combat them.

We surveyed 150 developer and DevOps managers who lead teams that mostly develop cloud-native applications (Kubernetes, Lambda, and more). We learned that although many of them release software very quickly and quite often, most still struggle with solving customer issues. In this report, we break down the source of these challenges and offer some insight as to how these struggles can be dealt with.

# Methodology

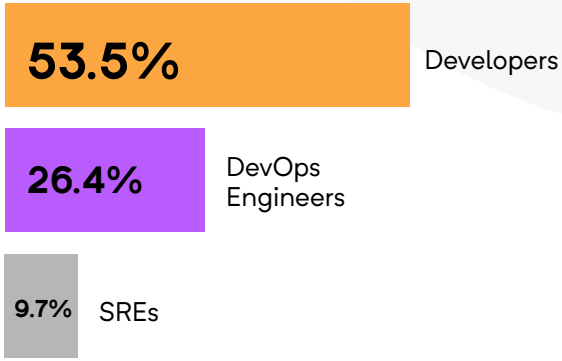
Over the course of 2020, Rookout conducted a poll of software engineering professionals, ranging from Tech Lead to Head of Engineering to CTO. Responses were gathered through a variety of channels, such as social media and third-party websites.

# Demographics:

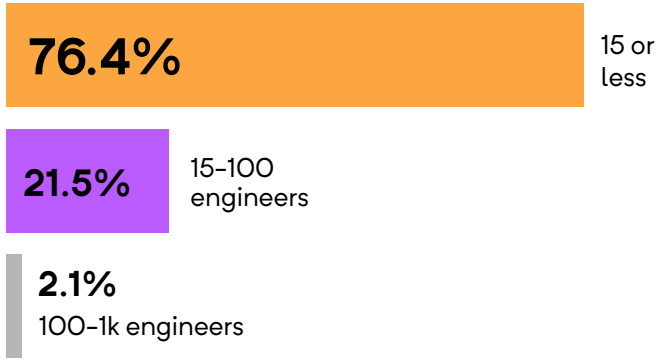
## Top Industries

- Connectivity solutions
- Medicine
- Security
- Car rental
- Universities
- Observability tools
- Music
- Aviation
- SaaS providers
- Food delivery
- Healthcare providers

## Type Of Team They Lead




## Team Size



CHALLENGE

1

# With The Rise Of Digital Transformation, Comes A Challenge In Maintaining Productivity And Velocity

The move towards microservices and serverless has negatively influenced software engineers' workflows. The difficulties they face when working with such environments have caused a decrease in productivity, velocity, and time to resolution of issues. Here is a more in-depth look at some of the challenges they face and why.

**70%**
of respondents are working with cloud-native or serverless.

**30%**
are working with monoliths.

The majority of respondents said that they are working with cloud-native technologies. They also reported that serverless, Kubernetes, and cloud-native are the technologies most frustrating for them to work with.

These technologies are very difficult to troubleshoot, due to their inherent nature and limitations.
As the adoption of these technologies is growing, it's important to note that these technologies make visibility and troubleshooting much more difficult, due to the ephemeral nature of the code.

## Discussion:

While cloud-native apps have their benefits – mainly in that they make it easier and cost-effective to scale up and scale out easily – they also bring a new set of challenges with them that may cause frustration in engineering teams attempting to adopt them.

The complexity they bring demands a steep learning curve, and one of their main value points – namely, the abstraction they provide between the application code and the hardware the application is running on – also raises new challenges when attempting to troubleshoot the application without access to the hardware they run on.

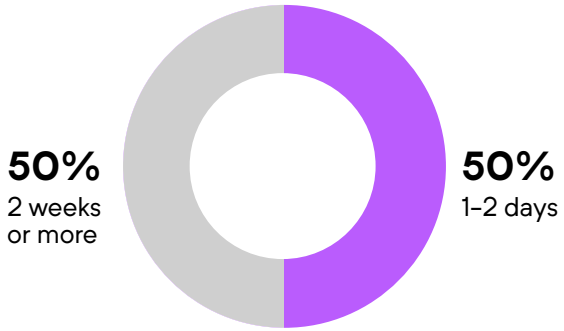
The market observability and understandability tools are starting to shift in an attempt to address these new challenges, but many software development organizations still struggle with developing their own tools and methods for troubleshooting cloud-native applications. In the next section, we will take a close look at how developer teams attempt to address these changes.

CHALLENGE

2

# Resolving Customer Issues In A Timely Manner

In every business, keeping up with customer requests and resolving customer issues is of the utmost importance.




Respondents reported that on average, it takes half of them two weeks or more to resolve customer issues. So where does that leave these respondents, and how do they make sure that this long period of time doesn't impact business results?

**70%**
of respondents also reported that on average, it takes their team on average anywhere from up to 2 weeks to a month to release and roll out a new feature.

**60%**
of respondents reported that they release changes to their product anywhere from once a week to once or twice a month. 30% reported that they release once a day. And how do they make deployment happen that often? Agile methodology.

## Discussion:

A large number of respondents have reported that it takes them a significant of time to resolve customer issues. However, the other half have said that they're able to solve customer issues within 1-2 days. One reason for this is the adoption of the shift-left methodology. Yet, when looking at it from the other side, a significant amount of the market, as reported in the survey, is still **taking two weeks or more to resolve customer issues**. And that's simply unacceptable.

CHALLENGE

3

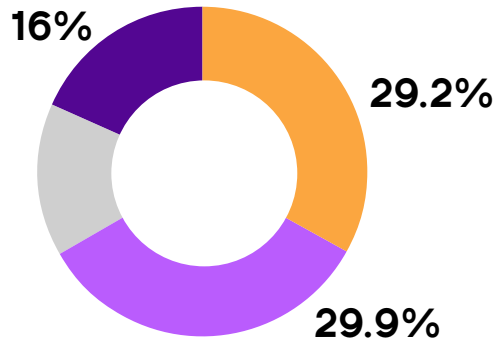
# Distributed Workforces Are Necessitating A Balance Between Speed And Quality

The rise of distributed workflows has been on an upwards trend for the past few years. The recent pandemic of COVID-19 has kicked it into overdrive. Our respondents have shown us that a key component that affects developer teams is the need to balance speed and quality.

Developers have the need to balance velocity and quality. It's a pretty simple equation: more speed causes more bugs, and slower speed means higher quality. When working distributed, this dilemma is even more emphasized, as both quality and speed can be negatively impacted.

When asked 'What are you measured by most as a leader?' respondents reported three key measurements:

1. Meeting release deadlines
2. Delivering features to make the customers happy
3. Finding as many bugs as possible as fast as possible




## Discussion:

Managers want to meet release deadlines on time, but the fast-paced nature of their team's work causes dissonance between the quality being delivered and the speed it's being delivered at. Delivering features to make customers happy is not just a business goal but another aspect of the shift-left trend in which engineers find themselves impacting business results more and more. Finding as many bugs as possible as fast as possible leaves developers with a crucial decision: do they fly blind or fly slow?

Essentially, all of these three measurements are related to speed and quality issues. That's the bottom line here. **73% of respondents reported that it's all about speed.**

CHALLENGE

4

# Distributed Workforces Are Hurting Teamwork And Collaboration

A distributed workforce comes with its own unique challenges, whether it's dev teams who sit far apart from each other, the use of communication channels like Slack instead of face-to-face meetings, understanding third-party code alone, or more.

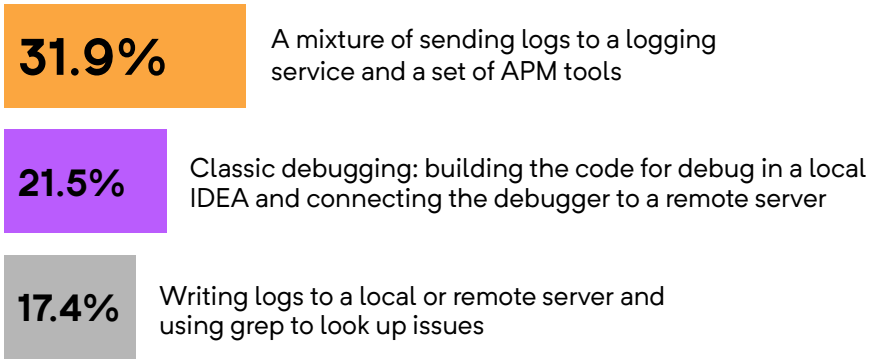
When respondents were asked, 'What is the greatest challenge holding you and your team back?', they reported that debugging and efficient coding were the biggest challenges. Second to those was the challenge of knowledge transfer and collaboration, with almost 20% of respondents reporting this.

We found that logging and APM tools dominate by almost 32%, followed by classic debugging and writing logs to a local remote server. These are all tedious processes, taking much time and resources to implement and run.

## Discussion:

Debugging and efficient coding are common dev team challenges and don't come as a surprise. The challenge that many respondents found in knowledge transfer and collaboration, however, was a surprise.

There is room for improvement here- both in terms of team culture and tools. Proper tools will help your devs resolve the speed vs quality issue and improve teamwork and collaboration when finding yourselves in a distributed workflow..



CHALLENGE

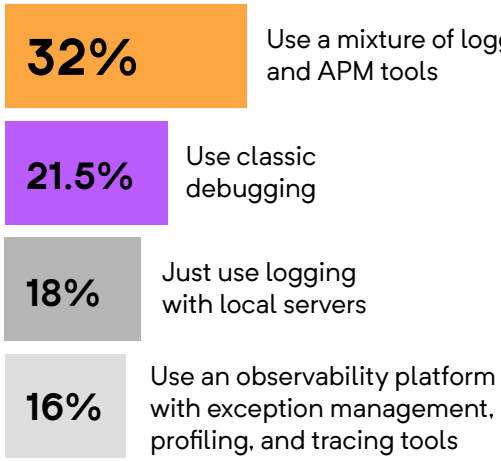
5

# Remote Debugging Is A Top Priority

With everything we've spoken about so far– the rise of digital transformation, the continuation of distributed workflows, and the need to resolve customer issues in a more efficient and timely manner– where do we find the state of debugging?

Many respondents reported that their top concern, and their biggest challenge holding them back, is: 25% efficient coding and feature quality and 18% debugging, reproducing bugs, and finding the source of the bug.

As it is, though, the majority of teams aren't using the right tools. They reported that, when debugging remote environments:




## Discussion:

As all developers know or have experienced, traditional debuggers simply don't work as efficiently with cloud-native technologies. Many developers continue to use them, whether by preference or due to a lack of proper tooling. By continuing to use traditional debuggers, they find themselves in a situation where they need to push through, go without the necessary data needed to debug, write a ton more logs, or a myriad of other options that are painful and time-consuming.

The solution is adopting the proper tool: a modern debugging solution that's built for cloud-native applications. Developers will be able to get the data they need from their code, no matter where it's running, and while their code is running live.

But these simply aren't enough.

45% of respondents reported that they or their teams debug 5-10 hours, and over 30% debug even more than that per week. What does this mean? Too much time spent debugging, instead of value-creating features. This goes against the main drive for digital transformation.

# About

Rookout empowers engineers to solve customer issues 5x faster, by making debugging easy and accessible in any environment; from cloud-native to on-prem and from dev to production. With its live data collection and debugging platform, Rookout allows software engineers to handle the complexity of modern applications by seeing into their code in real-time, as it's running. Using Non-Breaking Breakpoints, software engineers get the data they need instantly, without additional coding, restarts, or redeployment of their application required. See what Rookout can do for you at https://www.rookout.com