2024 Mobile Developer Pain Points

Survey results from over 1,000 app builders

embrace.io

embrace

Contents

Introduction

Mobile engineers are operating in an environment of both technological complexity and high consumer demands.

Mobile engineers today face several key challenges. They're building apps and games in the most technically complex and interconnected environments to date. At the same time, they're faced with the ever-increasing demands from their end users to make everything better, smoother, and faster.

ERIC FUTORAN
CEO & Founder

Achieving all of this in the hyper-competitive landscape of apps and games is no easy feat. It comes with its fair share of trials and frustrations. As a company built *by* engineers *for* engineers, we're well aware of this. We've experienced, first hand, so many of the challenges and pains that accompany the pursuit of building great software.

ANDREW TUNALL
Head of Product

Our tooling helps to mitigate some of those pains, specifically around performance. For us and our customers, this has been game-changing. But how critical is maintaining app performance in the wider world of mobile development, and especially in the greater scheme of engineers' many competing priorities? What other pains are they dealing with in their day-to-day? And how supported do they feel in addressing those pains with tools and other solutions?

To get to the bottom of these and other questions, we asked mobile engineers directly. We surveyed over a thousand engineers to understand their needs, motivations, and pain points in working in the high-demand world of mobile.

What we found validated our assumptions in some respects, while also surprising us in others.

Whether you're an individual contributor, engineering manager, VP, or even CTO, we believe the findings from this work can help inform your understanding of your team's experiences. Perhaps these findings may even sow some seeds of inspiration in how you can continue to support your teams in new ways as they take on the everyday challenges that come with building something great.

We hope you find this work as insightful as we have.

Key insights

01 Improving app performance is the top day-to-day priority

As part of their day-to-day work, mobile engineers prioritize improving the performance of the apps they support over releasing code with few/no errors, deploying new releases on time, and working on new and innovative features. This speaks to how crucial mobile performance is to the bigger picture, and how essential the right tools are to keeping performance sharp.

02 Spending too long fixing bugs is the no. 1 frustration

More than half of engineering ICs, as well as over 40% of engineering managers and lead/principal engineers, say that spending too long fixing bugs is a top frustration of their day-to-day work. This is despite the fact that 98% of respondents polled said they use some type of performance monitoring tool for mobile. This suggests that available solutions aren't doing enough to save time and reduce toil for engineers.

03 There's a gap in prioritizing tooling for IC engineers

Among individual contributors, accessing tools to make their work more efficient is among their top priorities. Additionally, ICs are the most likely of all engineering groups to say they're frustrated by not having the right tools to do their jobs. Not all levels of engineering management recognize this need, however. While 43% of engineering managers/team leads strongly believe that providing tools to make engineers' work more efficient is a top priority for their organizations, only 31% of senior leaders (VPs/SVPs/CTOs) think so. This suggests a disconnect that may be preventing ICs from getting the tools they need to work most efficiently.

Key insights

04 Good documentation, ecosystem support, and the ability to replicate user issues are paramount for performance monitoring tools

Among all engineering professionals, these three features are key when it comes to choosing a mobile performance monitoring solution. IC engineers and senior engineering leaders (VPs/SVPs/CTOs), while at opposite ends of the hierarchy, are strongly aligned on how important some of these features are. Both want solutions to integrate seamlessly with their org's other tooling. Both also want a tool that surfaces intelligent insights to help solve problems.

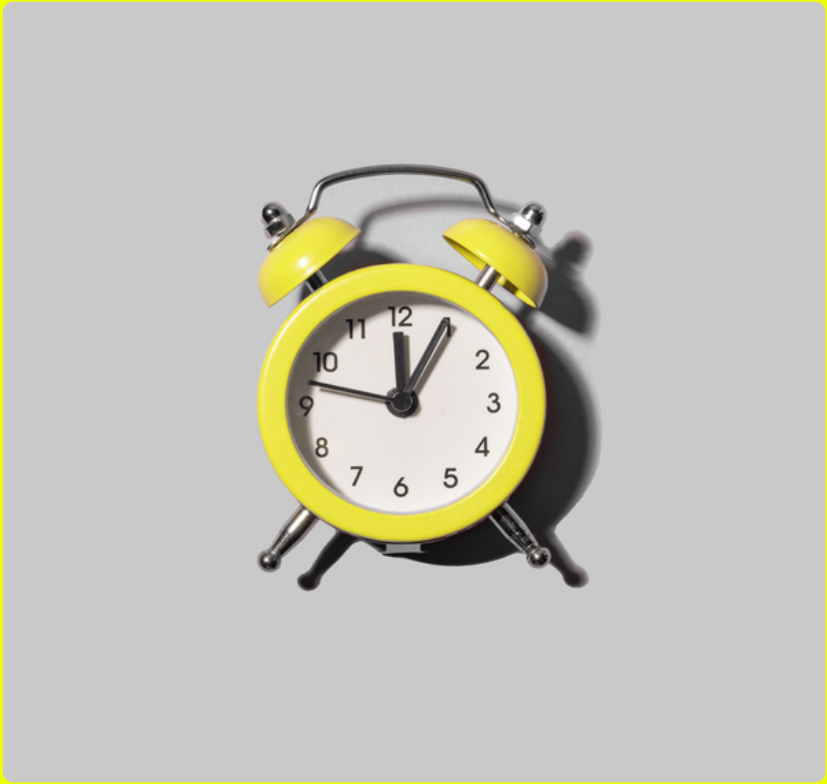
05 When it comes to integrating an SDK, team consensus is just as important as managerial approval

Nearly half of engineers say they need approval from a department head in order to integrate a new tool into production-level code, making it the most common hurdle to adopting a new SDK. However, 43% also said they need consensus across their team to do so, making consensus more important than approval from their direct manager, CTO, or legal advisor. This has implications for how those who provide SDK solutions enable and empower all members of a team to see the value in their tooling in order to gain trust and build advocacy.

06 Android and iOS engineers have different views which reflect the nature of their respective open vs. closed systems

Android engineers are generally more concerned with improving app performance and releasing error-free code as part of their day-to-day, which is understandable as the diversity of devices in the Android world make this more difficult. iOS engineers, on the other hand, are more likely to see automation and efficiency in their work as important. When it comes to performance monitoring tools, Android engineers prioritize tools that save them time resolving errors, while their iOS counterparts are more likely to want tools that integrate well within their existing ecosystem, as well as tools that offer dedicated customer support.

1 Day-to-day priorities

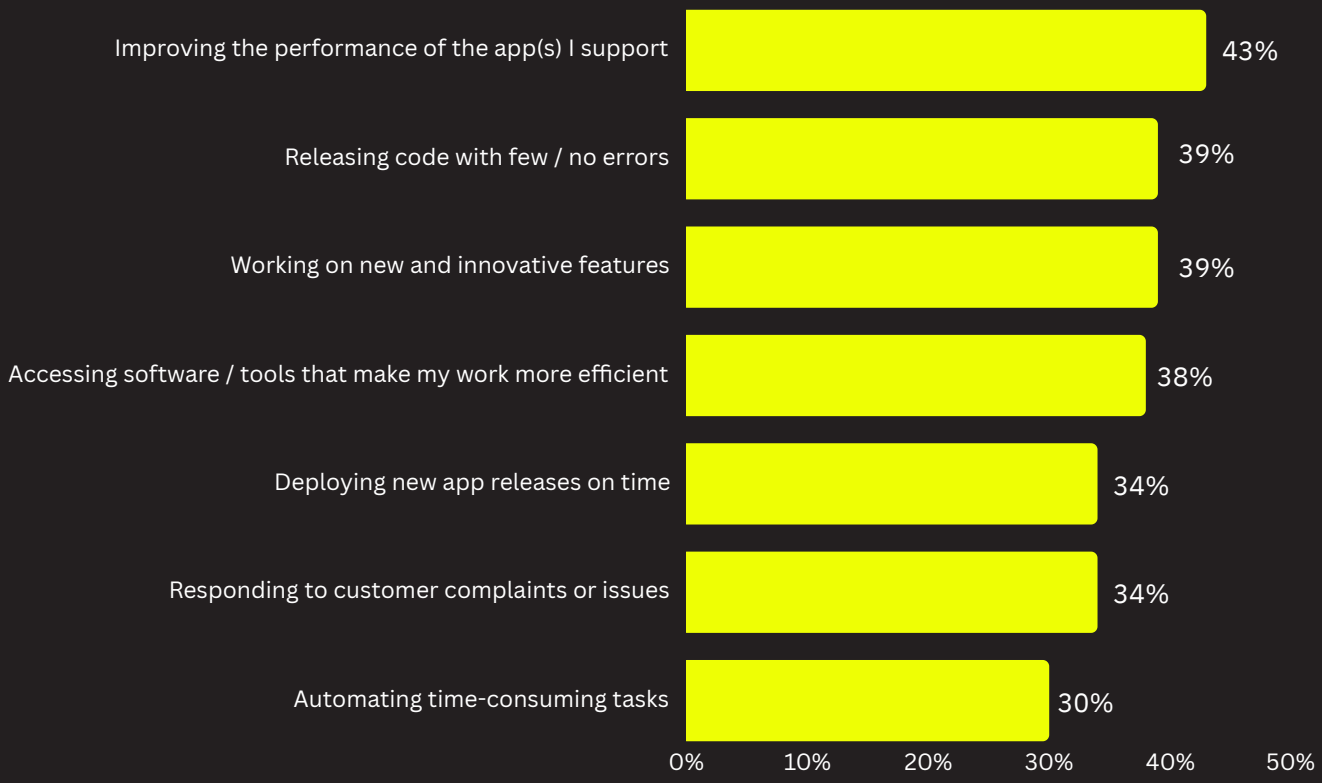


Improving app performance tops day-to-day priorities

Among all mobile engineering professionals we surveyed – a group which includes individual contributors, managers, and leaders – most respondents said improving the performance of the apps they support was the most important aspect of their day-to-day work. This was followed by releasing code with few or no errors, working on new features, and accessing tools to make their work easier.

The importance of improving performance, a trend we've seen from past research, stresses how critical monitoring tools are for engineers to do their best work, something we'll explore more deeply later in this report.

Q: Consider the following aspects of your day-to-day work. Which of these are most important to you personally? (Top 3)

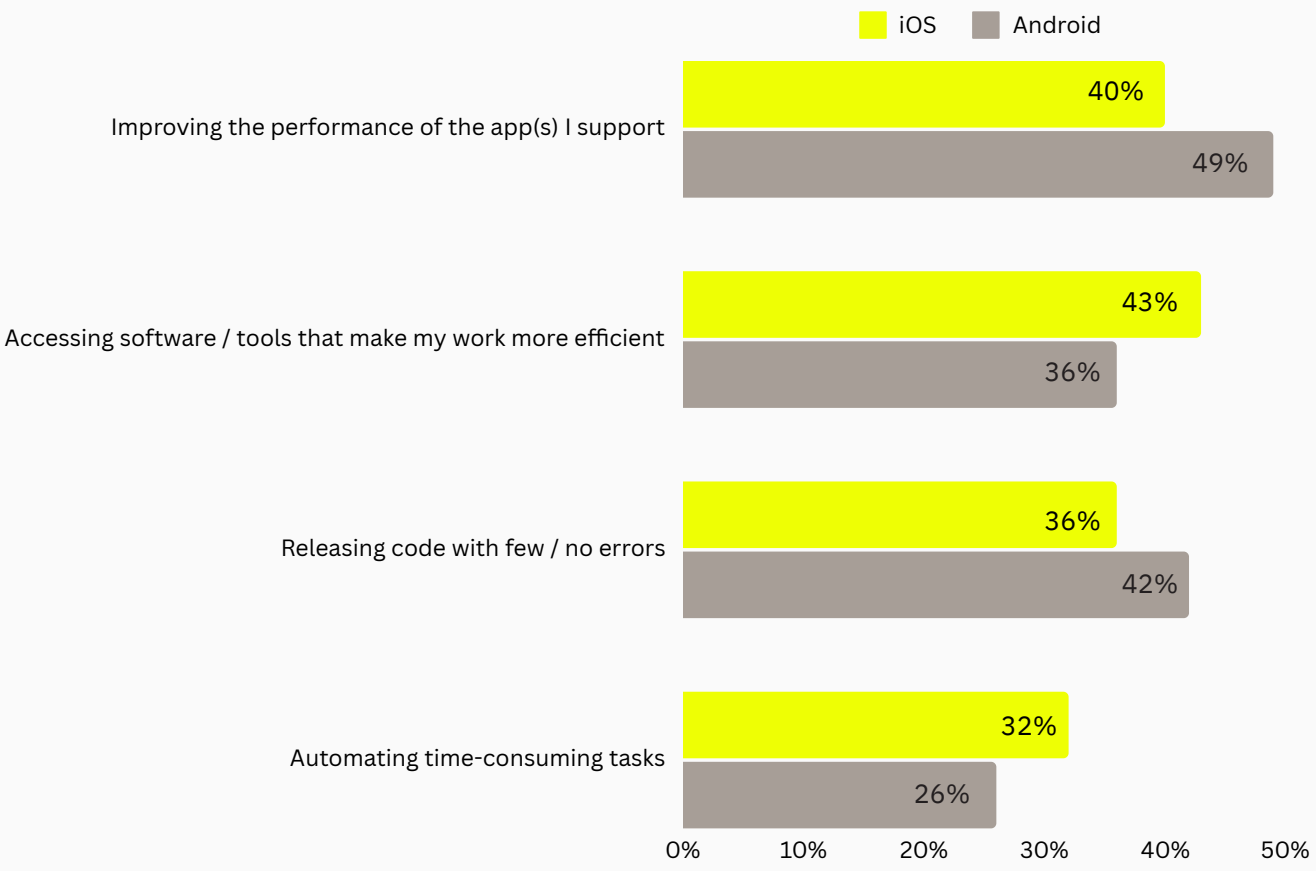


Efficiency and automation lead iOS priorities, while performance is paramount for Android

Examining these results by platform, we see that engineers who work exclusively on iOS tend to be more concerned with efficiency and automation in their workflows vs. those who work exclusively on Android. For Android respondents, improving performance and releasing code with few/no errors is paramount.

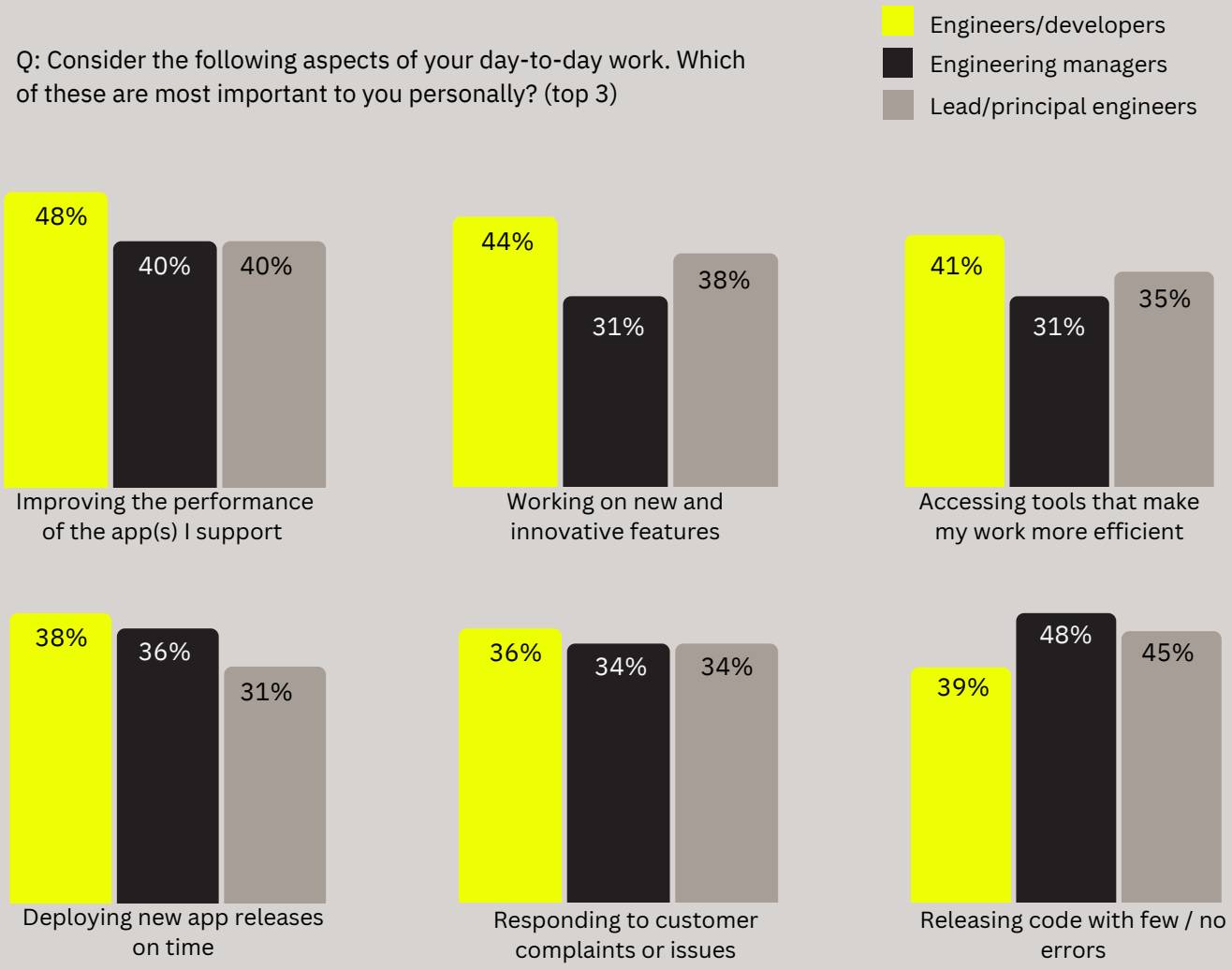
This is likely influenced by the extremely varied, open source nature of the Android ecosystem. With thousands of different devices (many of them older) and numerous versions of the operating system out in circulation, maintaining consistently strong performance is a greater challenge for Android engineers. In contrast, iOS is a much more regulated system.

Q: Consider the following aspects of your day-to-day work. Which of these are most important to you personally? (Top 3)



Importance of these priorities varies across the engineering hierarchy

Q: Consider the following aspects of your day-to-day work. Which of these are most important to you personally? (top 3)

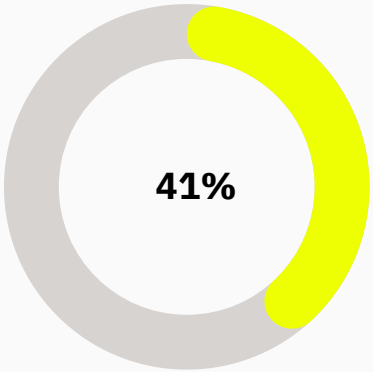


For engineers/developers, who are typically individual contributors within their organizations, there's an even stronger focus on improving the quality of the apps they work on, whether that's via improving performance or working on new and innovative features. Additionally, individual contributors at a lower level of the engineering hierarchy are more concerned with getting tools that make their work more efficient. Engineering managers and lead/principal engineers, on the other hand, prioritize quality control – for example, 48% of managers, compared to 39% of engineers, say that releasing error-free code is a top aspect in their day-to-day work. This is likely a reflection of the accountability that engineering managers have to broader company goals, such as ensuring service level agreements (SLAs) for their customers are met.

Clear gap in prioritizing tooling for individual contributors

Q: Which of these are most important to you personally?

Accessing tools that make my work more efficient



Engineers/developers

More than 40% of engineers say that accessing tools to make their work more efficient is a top priority for them.

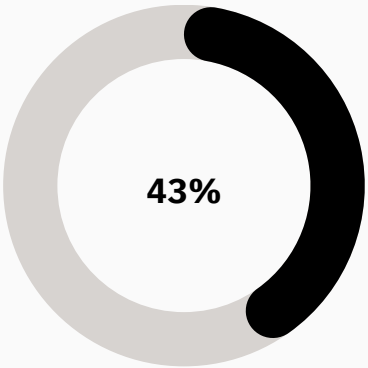
When various levels of engineering management were asked how important it is, from the perspective of their companies, to actually *provide* engineers with these tools, answers ranged widely.

Engineering managers, who generally have the closest working relationship with individual contributors, believe this is a high organizational priority.

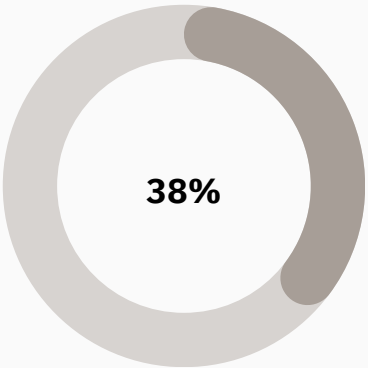
However, only 31% of VPs, SVPs, and CTOs considered providing tooling to make engineers' work more efficient as an important aspect of day-to-day work from a company perspective.

Q: Which would you say are most important from the perspective of your company / org?

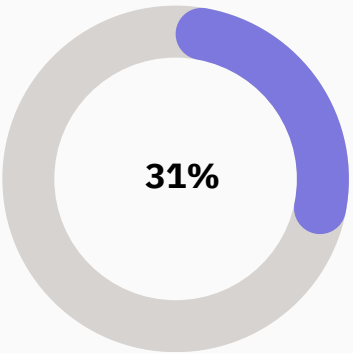
Providing engineers with tools that make their work more efficient



Engineering managers

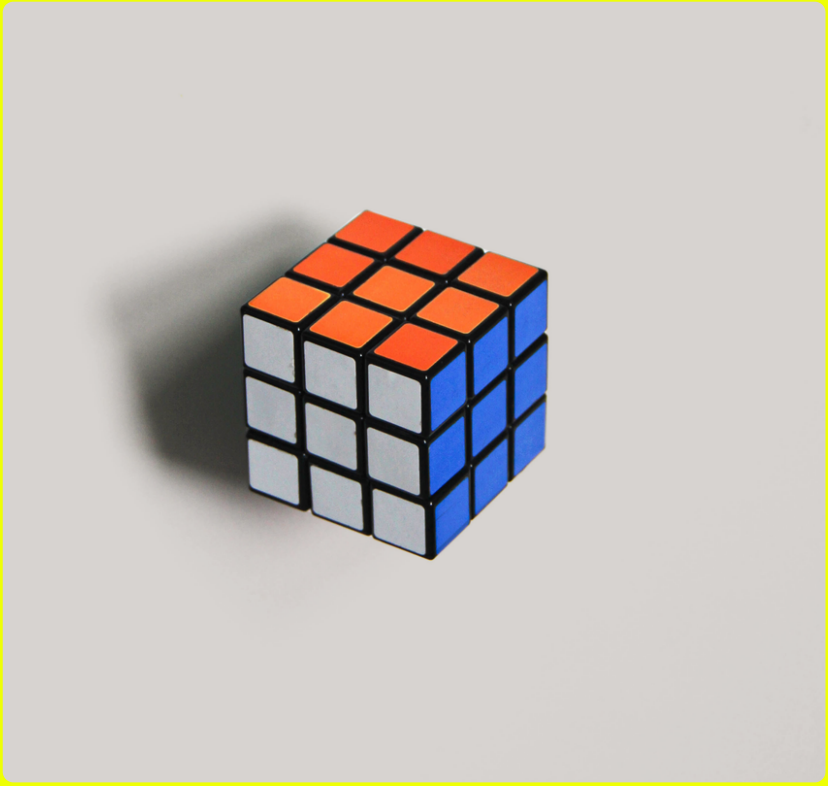


Lead/principal engineers



VPs, SVPs, CTOs

2 Day-to-day frustrations



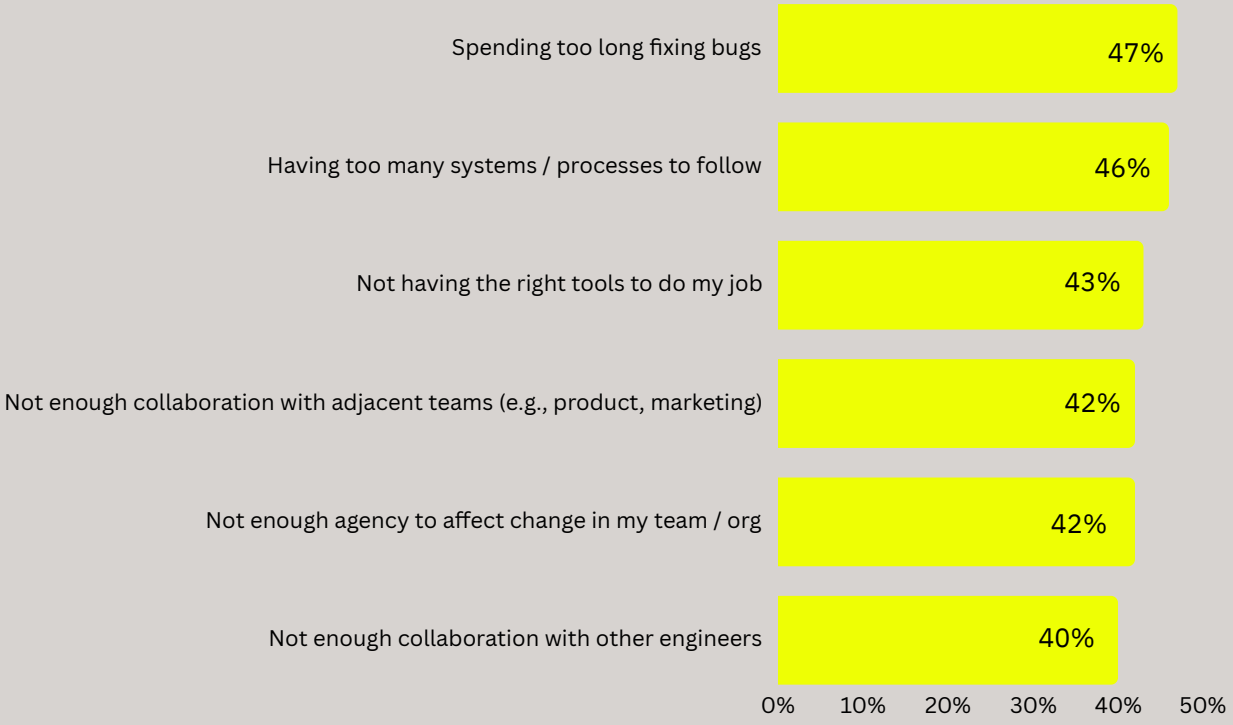
Time spent debugging tops list of frustrations

When asked what their biggest day-to-day frustrations were, mobile engineers reported that wasting time fixing bugs was the number one issue. This was followed by having too many systems/processes to follow and not having the right tools to do their jobs.

Answers among iOS vs. Android engineers did not differ hugely, though there were a couple of notable points that are consistent with other findings in this report. Spending too long fixing bugs was more of an issue for Android engineers, reflecting the greater challenge they face in maintaining consistently strong performance within a hugely varied ecosystem.

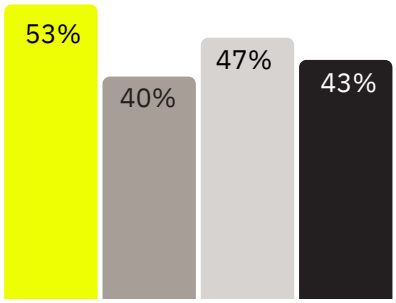
About 47% of Android engineers, compared with 42% of iOS engineers, also said they were frustrated by a lack of agency to affect change in their orgs.

Q: Which of these frustrate you the most in your day-to-day work? (Top 3)

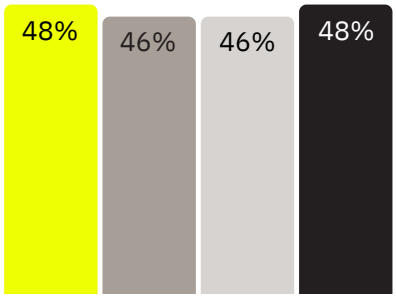


Key frustrations vary between ICs and Managers

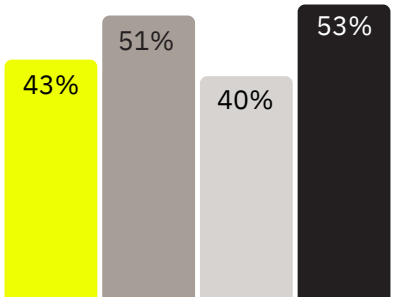
Q: Which of these frustrate you the most in your day-to-day work? (Top 3)



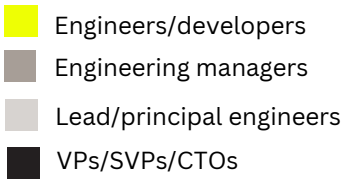
Spending too long fixing bugs



Having too many systems / processes to follow



Not enough collaboration with adjacent teams (e.g., product, marketing)



Among individual contributors, the primary frustration is by far spending too long fixing bugs.

Engineering managers and senior engineering leadership, on the other hand, are more likely to get frustrated by a lack of collaboration with adjacent teams, such as Product and Marketing.

Considering the different responsibilities of individual contributor vs. management roles, this is not surprising.

Having too many systems/processes to follow was reported as the No. 2 most frustrating aspect of day-to-day work among all respondents. Professionals across the spectrum of roles and hierarchies in engineering are equally likely to report on this frustration. In itself, this is an interesting observation into how process-driven engineering work has become today.

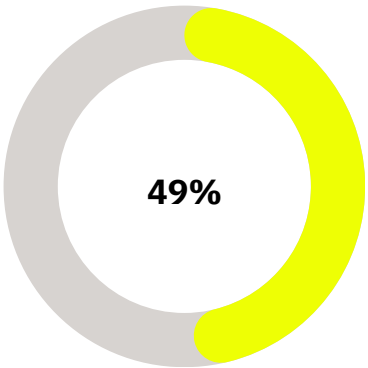
Lack of tooling drives frustration among ICs

Engineers who are individual contributors are the most frustrated out of all our respondents by not having the right tools to do their jobs.

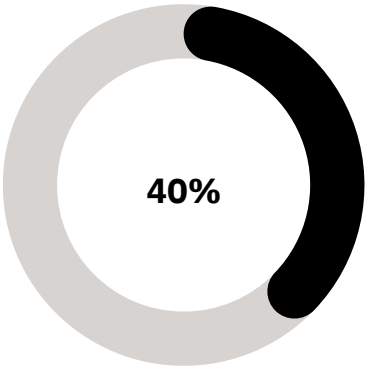
This reinforces some of our earlier findings that, while accessing tools to make engineers' jobs more efficient is very important to them, it's not always seen as critical as perhaps it should be by upper levels of management in engineering, leading to potential frustrations.

Q: Which of these frustrate you the most in your day-to-day work?

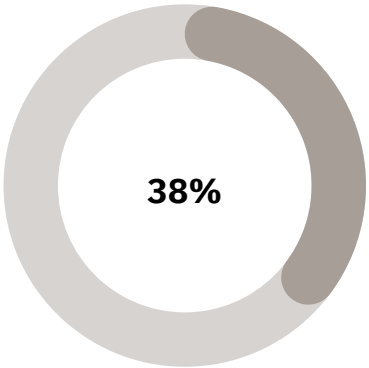
Not having the right tools to do my job



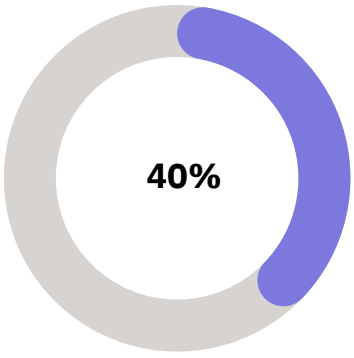
Engineers/developers



Engineering managers



Lead/principal engineers



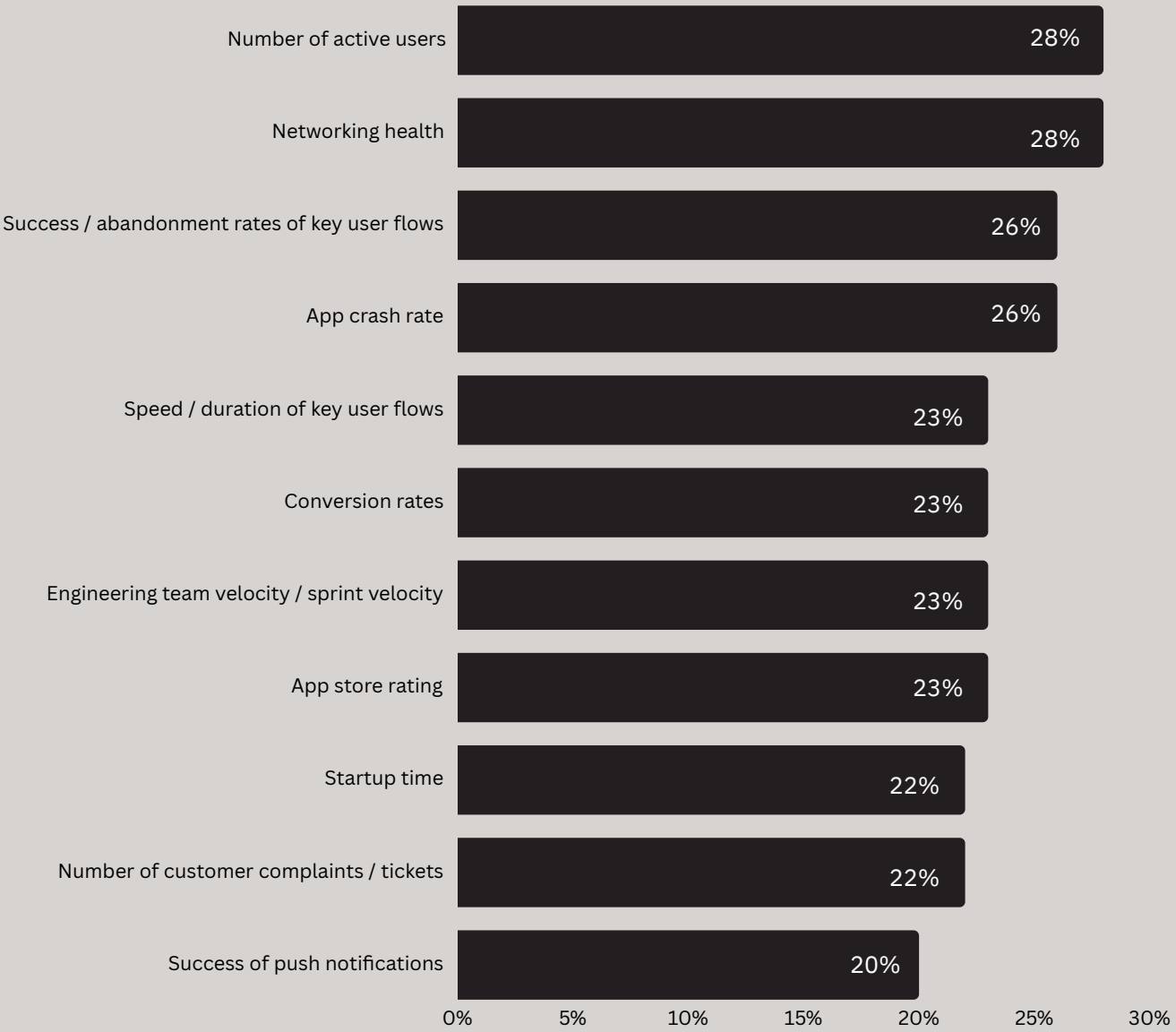
VPs, SVPs, CTOs

3 High-level mobile KPIs

Active user numbers and networking health top mobile engineering KPIs

When it comes to overall KPIs that engineering teams track, respondents mentioned active user numbers, networking health, success/abandoment rate of key flows, and crash rates as the top few metrics.

Q: What are the top 3 metrics or KPIs that your engineering team tracks?

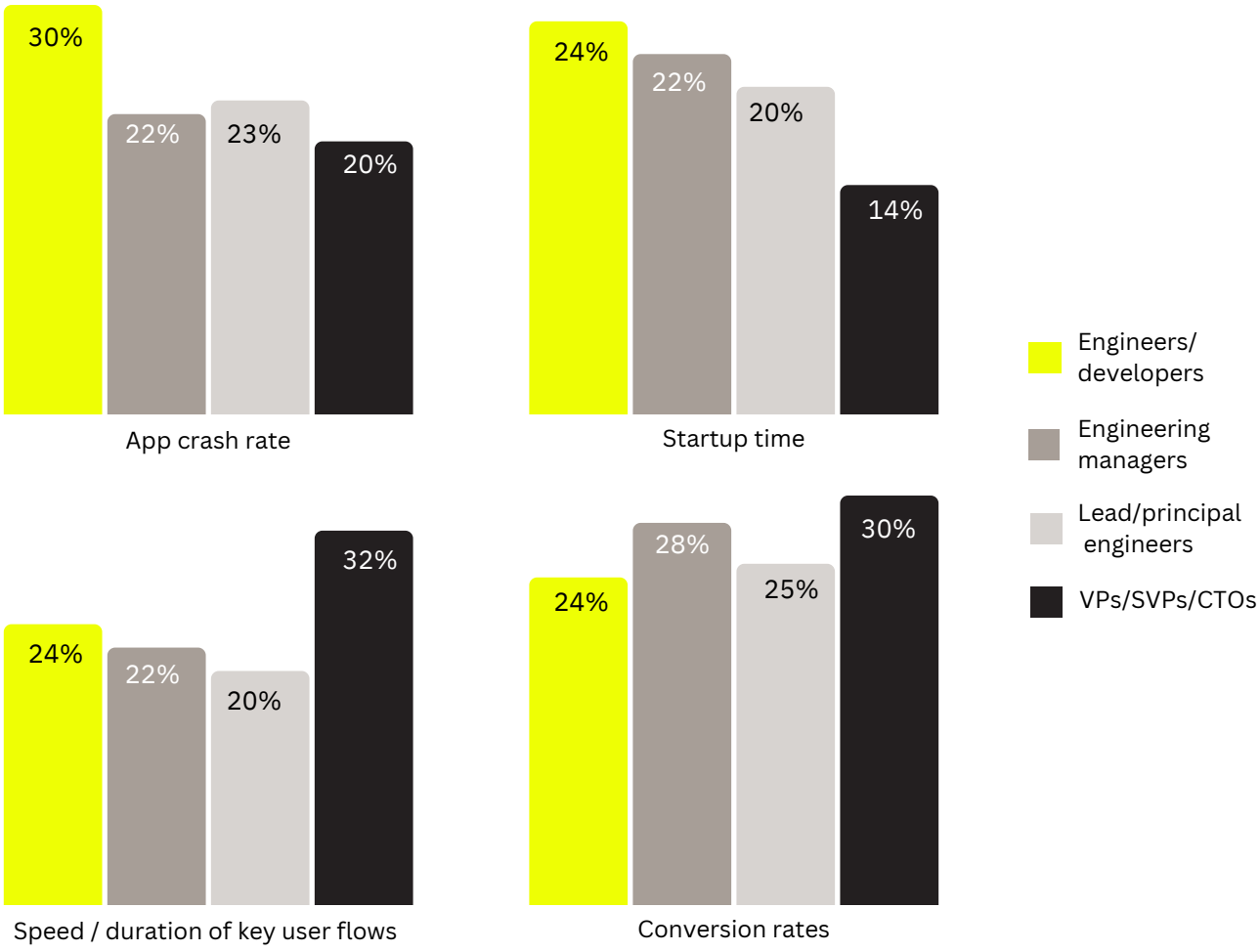


ICs more likely to track technical KPIs

There is considerable variation across the engineering hierarchy when it comes to the top mobile KPIs respondents said their teams track. IC engineers are more likely to say that highly technical KPIs, like app crash rate and startup time, are among their team's most important metrics.

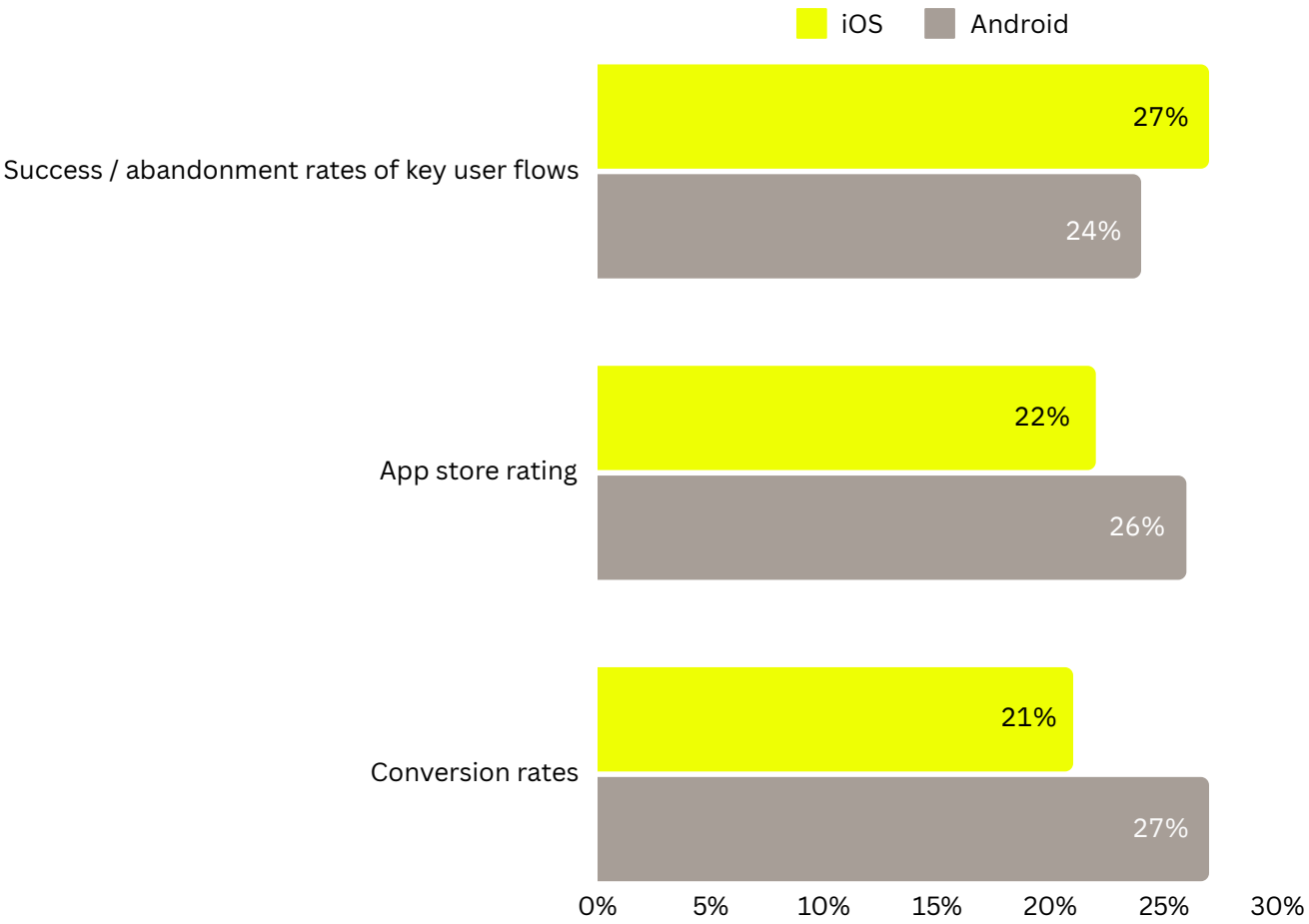
Engineering leaders like CTOs, VPs, and SVPs, on the other hand, are more concerned than other groups with commercial metrics, like conversion rates and the speed of key user flows (i.e. "check out" or "add to cart").

Q: What are the top 3 metrics or KPIs that your engineering team tracks?



App store ranking, conversion rates top-of-mind for Android

Q: What are the top 3 metrics or KPIs that your engineering team tracks?



Top engineering KPIs tracked across iOS and Android respondents varied only slightly. Android engineers were more likely to report that conversion rates were among their most important metrics tracked (27% vs 21% for iOS), as were app store ratings (26% vs. 22% for iOS).

Android apps are sensitive to the fluctuations of the Google Play Store ranking system because they are susceptible to ANR (Application Not Responding) errors. The Play Store punishes Android apps whose ANR or crash rates are above a certain threshold as this negatively impacts the user experience. This is likely why we see a greater focus on app store ranking among the Android group.

4 Perspectives on tooling

Despite prevalence of tools, data limitations pain engineers

A full 98% of professionals we surveyed reported they are using some sort of tool to monitor their mobile app's performance. This makes sense, as the No. 1 most important day-to-day activity reported by engineers was to improve the performance of the apps they support.

What's surprising is that, despite near-universal adoption of tools, the top daily frustration among engineers is spending too long fixing bugs. This suggests that existing tools are simply not doing enough to make people's lives easier.

One reason for this may be the diagnostic limitations of free and low-cost solutions. While free tools are among the most popular, engineers who use them report a number of complaints. At the top of this list are free tools' lack of capabilities to solve speciality issues (like ANRs and networking issues) as well as their data collection limits (sampling, rate limiting, etc.).

While free tools have a place, relying on them too heavily seems to come at a cost.

Key complaints with free/low-cost tools

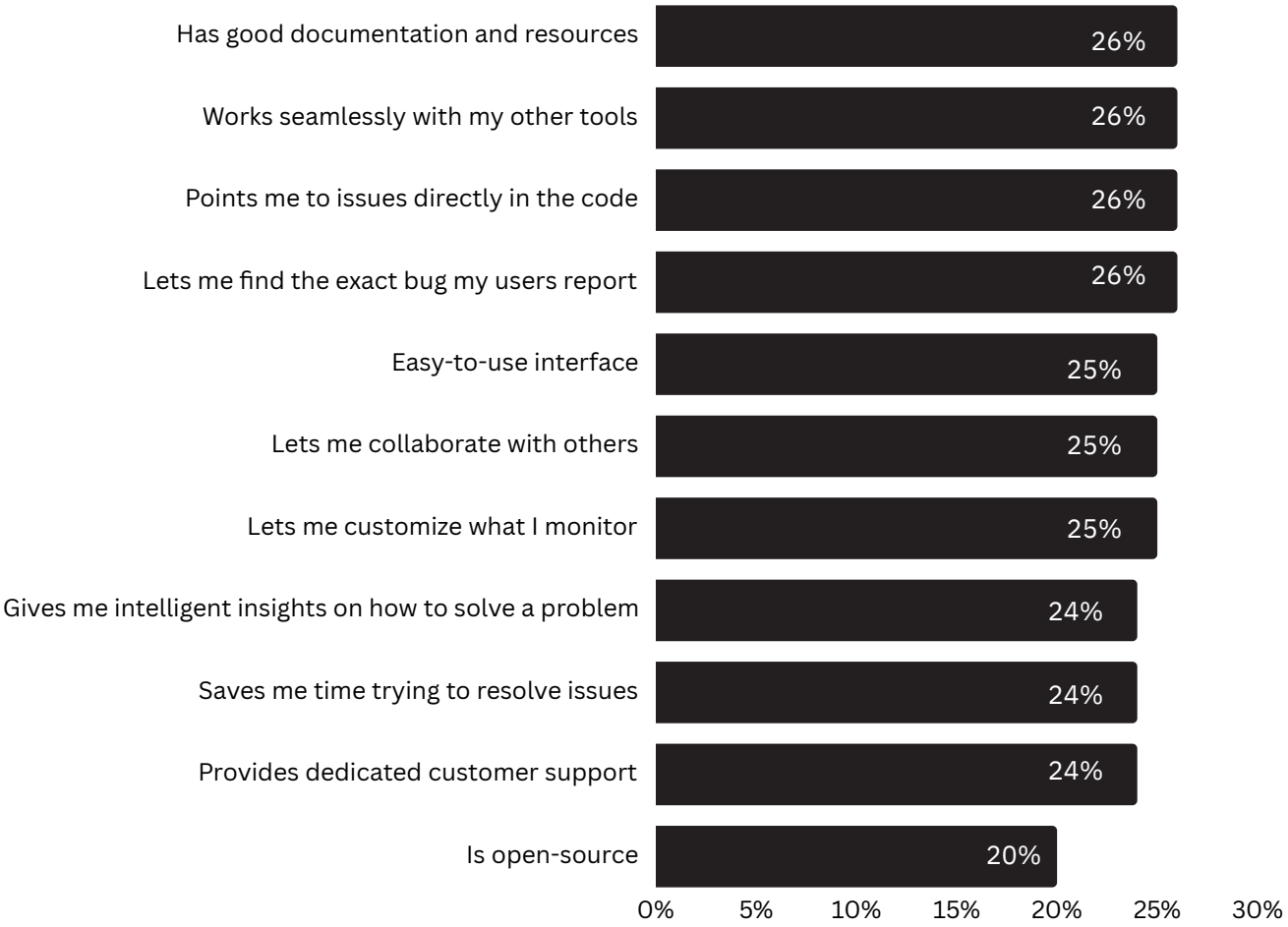
- Not enough functionality for speciality issues (ANRs, networking issues)
- Data limitations (sampling, rate limiting)
- Not compatible with other tools my org uses (DevOps, CI/CD, analytics)
- Data privacy concerns
- Not enough detail/granularity of data

Lack of consensus on importance of features for monitoring tools

Asking mobile engineers about how important different features and qualities are when it comes to a performance monitoring tool yielded highly varied responses. Having good documentation, working seamlessly with other tools, pointing directly to issues in their code, and letting engineers find the exact bug their users report were at the top.

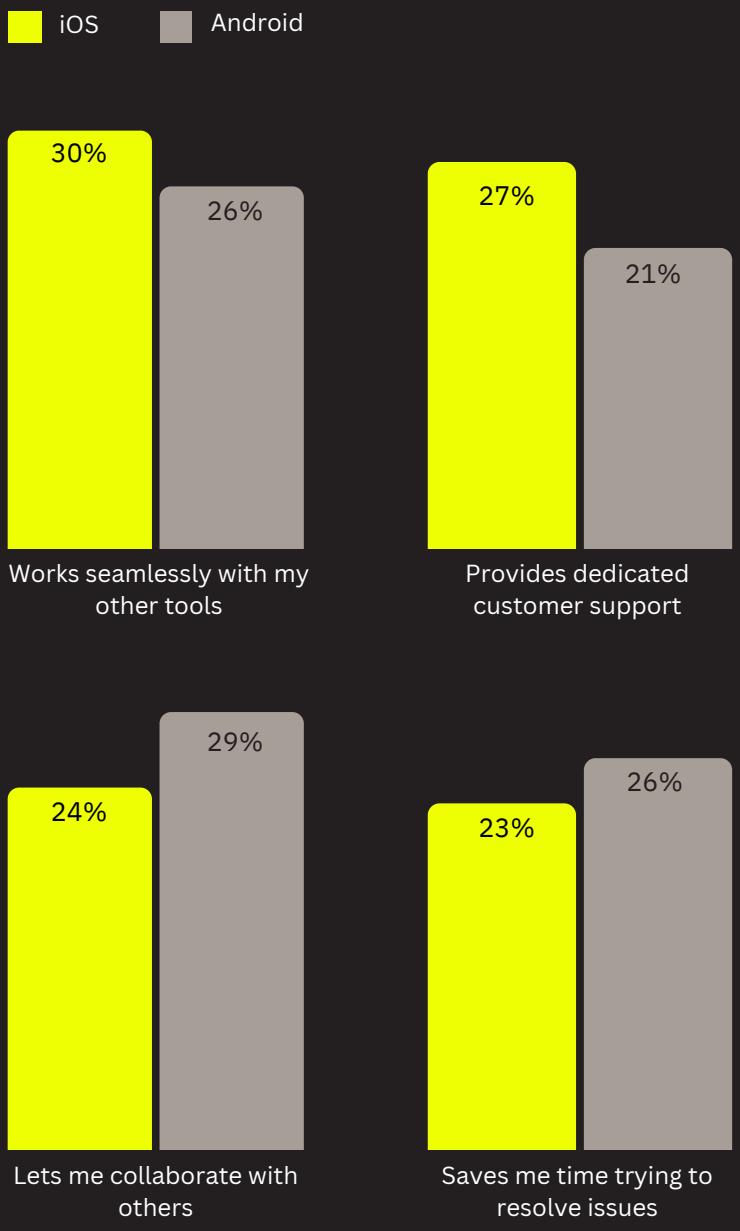
Feature preferences varied quite a bit across iOS vs. Android engineers, as well as across the hierarchy of engineering roles.

Q: When it comes to a performance monitoring tool for mobile, what's most important to you? (Top 3)



Where iOS and Android engineering priorities differ

Q: When it comes to a performance monitoring tool for mobile, what's most important to you? (Top 3)



Among iOS engineers, having a performance monitoring tool that works seamlessly with their other tools was more important vs. for Android engineers. Considering Apple's ecosystem is not as open as Android's, this makes sense.

iOS engineers were also more likely to value a tool that provides dedicated customer support.

Android engineers, on the other hand, were more likely to report that time-savings for issue resolution and collaboration were important features for them in a monitoring tool.

Common views between ICs and senior leaders

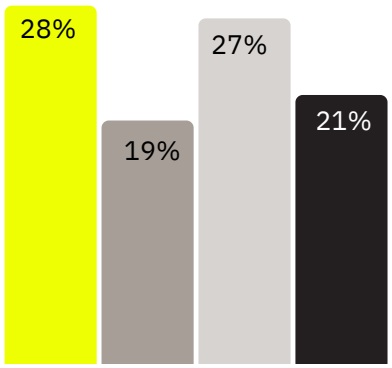
Across the many features of performance tools for mobile, their importance varies widely depending on one's role within the engineering hierarchy.

Individual contributors are more concerned with tools that can help them find the exact bug their users are reporting, as they are the ones responsible for troubleshooting the code. These engineers, along with their managers, are also very concerned with finding tools that can save them time during issue resolution.

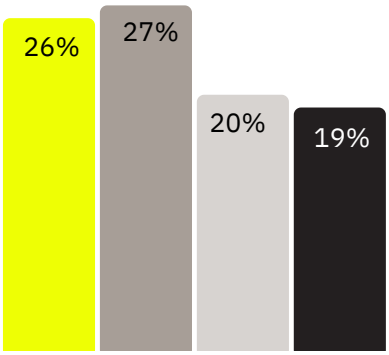
We also observed some commonalities between engineer ICs and senior leadership. Both ICs and VPs/SVPs/CTOs over-indexed in their desire for solutions that work seamlessly with their other tools, as well as for solutions that provide them with insight for problem-solving.

This common ground can be useful when thinking about how to communicate solutions that appeal to both the ICs who will be using them, and the engineering leaders who will ultimately be making a purchasing decision.

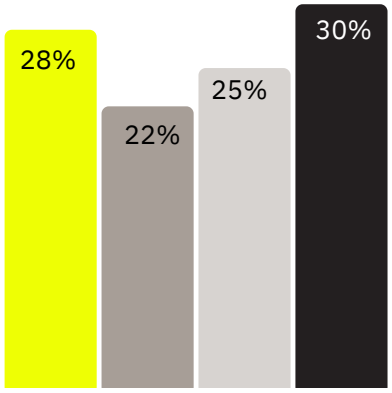
Q: When it comes to a performance monitoring tool for mobile, what's most important to you? (Top 3)



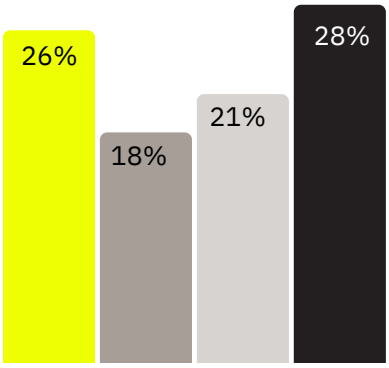
Lets me find the exact bug my users report



Saves me time trying to resolve issues



Works seamlessly with my other tools



Gives me intelligent insights on how to solve a problem

- Engineers/developers
- Engineering managers
- Lead/principal engineers
- VPs/SVPs/CTOs

5 Purchasing and decision-making

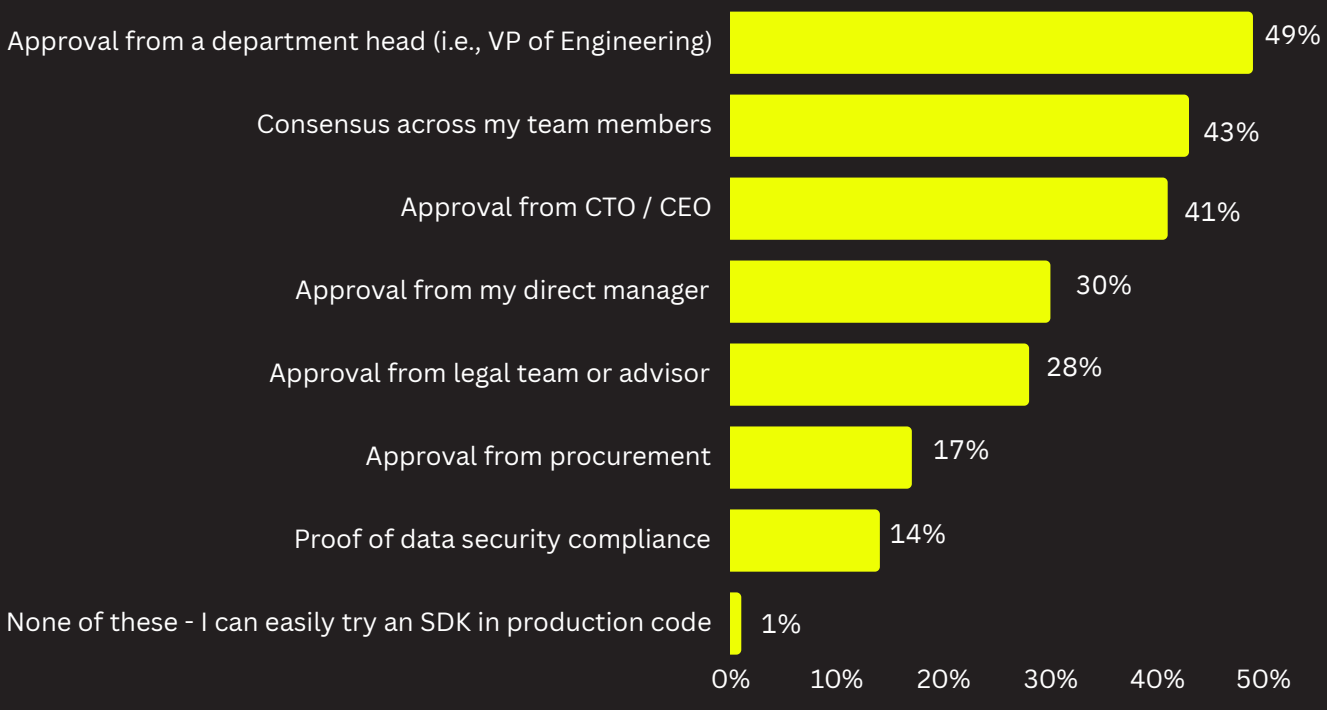
Team consensus is as important as approval

When it comes to actually implementing new tools that require integration into production code (as most SDKs used for performance monitoring do), a stringent approval process is typical. Half of respondents we surveyed said they would need approval from a department head, such as a VP of Engineering, to try an SDK in production code. About 41% actually said approval from a CTO or CEO would be necessary. Only 1% of engineering professionals felt confident that they could try an SDK in production themselves without going through an approvals process.

Interestingly, 43% of respondents said that consensus across their team members was required to try an SDK in production, making it the second most frequently named requirement to do so, just under department head approval.

For software providers, this suggests that making our tools accessible and attractive to entire teams, such as during a supported trials phase, is very important in order to garner shared consensus and advocacy across the team.

Q: What's required for you/your team to try an SDK in production code?



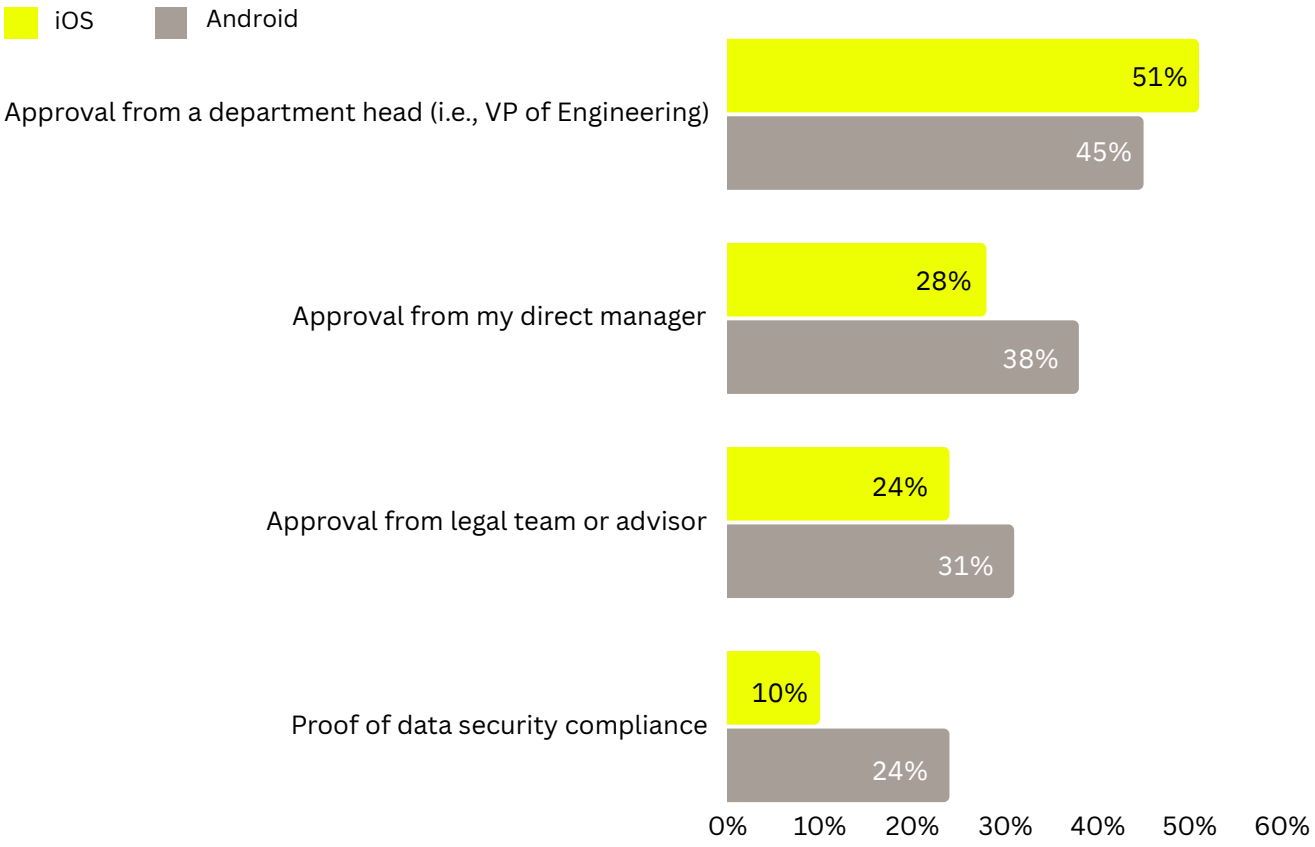
For some forms of approval, iOS and Android differ

Differences emerge between iOS and Android engineers when it comes to how much approval – and *whose* approval – is needed to add an SDK to production code.

While iOS engineers are more likely to need approval form a department head, Android engineers are more inclined to say that a direct managers' approval is needed.

Additionally, there's an added element of data security and legal compliance that Android respondents say they need. About 24% of Android engineers, compared with only 10% of their iOS counterparts, say they would need proof of data security compliance to implement an SDK in production. This is likely related to Apple's highly regulated "privacy by design" approach to its OS and any ancillary services. Apple already enforces a pretty robust set of privacy requirements for third-party SDKs, likely making ad-hoc data security reviews unnecessary.

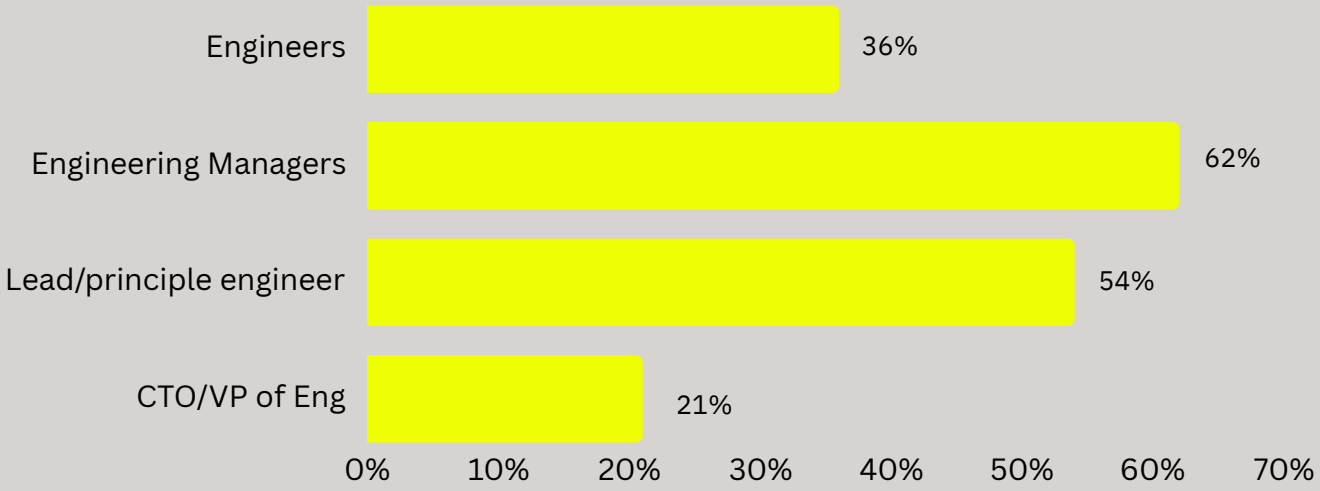
Q: What's required for you / your team to try an SDK in production code?



Engineering managers are lead decision-makers

Q: How much decision-making influence do you have when it comes to buying software tools for you/your team to use as part of your work?

% who say “a lot” or “full decision-making influence”



While CTOs and VPs of Engineering might have to ultimately approve tools for SDK integration, they don’t see themselves as being too influential in the decision-making process when it comes to finding and buying tools.

Respondents at the top and bottom of the engineering hierarchies have yet another thing in common when it comes to purchasing influence, in that both groups feel they have the least amount of it. About 36% of IC engineers said they have “a lot” or “full decision-making influence” when it comes to buying software for their teams, with CTOs/VPs of Engineering being even lower – at 21%.

On the other hand, 62% of engineering managers and 54% of lead/principal engineers reported having a substantial level, if not full, purchasing influence.

For tooling providers, this opens up the question as to who truly is the decision-maker they should be targeting as part of a marketing effort or sales process. Department heads and executives may be ultimate approvals for anything that will impact production code, but team leads and managers seem to hold greater sway when it comes to picking the right tools.

Final takeaways

Among the mobile engineers we talked to, there's a clear finding that achieving and maintaining strong app performance is a major driving force in their day-to-day work. Our research also suggests that, despite a competitive tools market, this need is not being fully met. After all, the No. 1 frustration engineers report is that they're spending too much time fixing bugs. This is despite the fact that 98% of them are using performance monitoring tools. So how can we better address the pains of mobile engineers?

Understanding ICs' perspectives is crucial, as they most acutely experience the pain of not having the right tools

Engineers who are individual contributors are the most likely to say they're frustrated by not having the right tools to do their jobs more efficiently. And while managers have the most say in actually purchasing these tools for their teams, ICs do have indirect influence in the form of group consensus. A huge number of respondents we talked to report that team consensus is important – more important, in fact, than most other types of approval – in determining what can be put into production-level code. Engineering leaders and organizations as a whole can benefit from getting the input of ICs earlier in the process when making decisions that will impact day-to-day development work and app maintenance.

iOS and Android engineers need tools that cater to their unique needs and challenges

There is no "one size fits all" solution for the entirety of the mobile development space when it comes to performance monitoring, and engineers know this well. The needs and frustrations of iOS and Android devs differ, in large part due to how these platforms are set up as closed/highly regulated vs. open/diverse operating systems.

To keep up with the complexities of mobile, engineers need monitoring tools that provide depth, breadth, and insight

Mobile infrastructure has become very complex and highly interconnected. For engineers, it's no longer enough for tools to just save them time on debugging their apps. They must do *more*. Engineers need solutions that can provide both depth and breadth by pointing to issues directly in code, as well as broadly integrating with all the other tools in their ecosystem. And, as AI becomes more commonplace in tooling, engineers will seek solutions that can provide intelligent insight into the issues they need to resolve. These are the tools that will truly free devs to do creative engineering work and less code maintenance.

Methodology

The data in this report comes from a survey of 1,116 mobile engineering professionals globally.

The approximately 10-minute survey was conducted online. Respondents were recruited via Embrace's social media channels, sponsored placements in engineering newsletters, and direct outreach. In order to qualify for the survey, respondents were required to hold any of the following titles: engineer/developer, engineering manager, principal/lead engineer, software architect, VP/SVP of engineering, or CTO. Additionally, respondents were required to work on either a consumer mobile app or internal company mobile app.

Respondent profile

Engineer/ developer	578
Lead engineer	172
Software architect	125
Principal engineer	93
VP/SVP of engineering	78
Engineering manager	67
CTO	3

iOS (exclusive)	595
Android (exclusive)	179

Consumer app	540
Consumer app AND internal app	244
Internal app	332

Embrace helps mobile engineers identify and prioritize user-impacting app issues with the detailed technical context to resolve them instantly. We provide full user session insight to solve any issue, whether it's a crash, slowdown, ANR or something else. When users are finding problems first, and mobile teams are finding out last, using Embrace empowers quick, early detection of issues before they become widespread. Engineers can focus on building the best player experiences possible and leave the fire drills behind.

Contact

8569 Higuera St, Culver City, CA 90232

(424)-326-9004

contact@embrace.io

embrace.io