APRIL 2024

STATE OF CONTINUOUS INTEGRATION & CONTINUOUS DELIVERY REPORT

The Evolution of Software Delivery Performance

Created by

/DATA CD.FOUNDATION

Sponsored by

CloudBees.

ABOUT THE CONTINUOUS DELIVERY FOUNDATION

The Continuous Delivery Foundation (CDF) serves as the vendor-neutral home of many of the fastest-growing projects for continuous integration/continuous delivery (CI/CD). It fosters vendor-neutral collaboration between the industry's top developers, end users and vendors to further CI/CD best practices and industry specifications. Its mission is to grow and sustain projects that are part of the broad and growing continuous delivery ecosystem.
For more information on the CDF or our projects please visit: https://cd.foundation/

CDF MEMBERS - PREMIER


CDF MEMBERS - GENERAL

CDF MEMBERS - END USER

CDF MEMBERS - ASSOCIATE

Can I share data from this report?

1. License Grant
This report is licensed under the Creative Commons Attribution-NoDerivatives Licence 4.0 (International) . Put simply, subject to the terms and conditions of this license, you are free to:

Share — You can reproduce the report or incorporate parts of the report into one or more documents or publications, for commercial and non-commercial purposes.

Under the following conditions:

Attribution — You must give appropriate credit to SlashData™ , and to the Continuous Delivery Foundation as sponsors of this report, and indicate if changes were made. In that case, you may do so in any reasonable manner, but not in any way that suggests that SlashData™ endorses you or your use.

NoDerivatives — you cannot remix or transform the content of the report. You may not distribute modified content.

2. Limitation of Liability
SlashData™, believes the statements contained in this publication to be based upon information that we consider reliable, but we do not represent that it is accurate or complete and it should not be relied upon as such. Opinions expressed are current opinions as of the date appearing in this publication only and the information, including the opinions contained herein, are subject to change without notice. Use of this publication by any third party for whatever purpose should not and does not absolve such third party from using due diligence in verifying the publication's contents. SlashData™ disclaims all implied warranties, including, without limitation, warranties of merchantability or fitness for a particular purpose.

SlashData™, its affiliates, and representatives shall have no liability for any direct, incidental, special, or consequential damages or lost profits, if any, suffered by any third party as a result of decisions made, or not made, or actions taken, or not taken, based on this publication.

The analyst of the developer economy | formerly known as VisionMobile

ABOUT THE AUTHORS

Liam Dodd

Senior Market Research Analyst

Liam is a former experimental antimatter physicist, and he obtained a PhD in Physics while working at CERN. He is interested in the changing landscape of cloud development, cybersecurity, and the relationship between technological developments and their impact on society.

liam.dodd@slashdata.co

Brayton Noll

Senior Market Research Analyst

Brayton Noll is a behavioral scientist with a background in climate change and environmental research. He holds a PhD from TU Delft in computational social-science with his thesis focusing on human behavioral dynamics and climate adaptation. He has five years of experience working with data analytics.

brayton.noll@slashdata.co

TABLE OF CONTENTS

KEY INSIGHTS

- DevOps adoption remains high, with 83% of developers involved in DevOps-related activities as of Q1 2024. →
- Developers with less experience in software development have a lower adoption of DevOps practices and technologies. →
- Continuous Integration and Continuous Delivery are the fourth and fifth most popular DevOps-related activities. →
- Source control management and issue tracking hold the top spots for the most widely used DevOps technologies. →
- While the proportion of top performers for lead-time for code change, deployment frequency, and time to restore service are down from when we began tracking, in Q3 2020, they now appear stable. →
- The proportion of low performers for each of the deployment performance metrics is increasing, which is a worrying trend. →
- Among developers at organizations with more than 1,000 employees, the deployment performance metrics are stable. →
- There is a strong correlation between the number of DevOps technologies used by developers and their likelihood of being a top performer across all three performance metrics tracked. →
- Using CI/CD tools is associated with better deployment performance across all DORA metrics, but is greatest among those simultaneously using both managed and self-hosted CI/CD tools. →
- Using multiple CI/CD tools, of the same form, leads to worse deployment performance, likely as a result of challenges related to interoperability. →

INTRODUCTION

01

1. Introduction

Continuous Integration (CI) and Continuous Delivery (CD), often combined and called CI/CD, are software development practices that enable developers to frequently integrate code changes and release frequent software updates reliably and safely. Closely linked to the broader DevOps cultural movement, CI/CD consists of a set of practices to automate and streamline the software delivery process. These practices, in turn, allow developer teams to innovate faster by collecting regular user feedback, prioritize the product features and fixes that matter most, and reduce risk.

This poses the question: to what extent – *really* – have developers embraced CI/CD practices and the DevOps culture to increase the effectiveness of their software development and release process?

In this report, commissioned by the Continuous Delivery Foundation (CDF) and authored by SlashData, we explore the current state of the CI/CD developer ecosystem and how this has evolved over the past three and half years. We will look at what proportion of developers are involved in the broader DevOps space and how many work with CI/CD technologies in particular.

Moreover, we will see how developers software delivery performance[1] has changed over time. Finally, we will explore how developers' usage of multiple CI/CD tools contributes to software delivery performance, as well as the impact of utilizing numerous different types of DevOps technologies.

The findings in this report are based on data from SlashData's previous eight Developer Nation surveys which reached more than 150,000 respondents worldwide over three and half years, from Q3 2020 to Q1 2024.

[1] *Lead-time for code change, deployment frequency, and time to restore service*

WHO IS INTO DEVOPS?

02

2. Who is into DevOps?

For the last eight consecutive iterations of SlashData's biannual Developer Nation survey, we asked developers whether they are involved in any of the activities that commonly fall under the DevOps spectrum, such as CI, CD, and testing applications for security vulnerabilities.

As of Q1 2024, 83% of developers are involved in DevOps-related activities. This highlights that developers are adopting DevOps practices in large swathes, even if not identifying as DevOps "specialists". While the current proportion of those involved is a small decrease from the peak Q1 2023 (85%), the decrease is mostly driven by newer developers. A quarter of developers with less than two years of experience in software development are not involved in any DevOps-related activities. This indicates that although the vast majority of new developers are engaging in DevOps activities, newer developers should be better educated about the benefits and usefulness of DevOps practices in maximizing skill development throughout their careers. Alternatively, it could also suggest that some organizations may want the more skilled/experienced developers working on software delivery processes. In summary, while DevOps involvement has decreased a small amount over the last year, it still higher than DevOps involvement in Q1 2022, when it was 77%.

We have seen a small decrease in the proportion of developers involved in all DevOps-related activities. While a small part of this can be attributed to the rise in developers not involved, the major driver is the continuous trend of developers consolidating around fewer activities over the past 18 months. This likely represents a maturity of developer and organization approaches to DevOps.

As DevOps grew in popularity and awareness, developers and organizations were likely engaging in a wide range of activities as they explored the benefits available from them. As we enter a more mature DevOps environment, the utility and effectiveness of activities are likely better understood. This means that developers are focusing on those most relevant to their roles, and organizations are likely dividing responsibilities among developers and teams. Further to this, aspects that were originally the responsibility of developers may have been automated, meaning fewer developers are required to manage these processes.

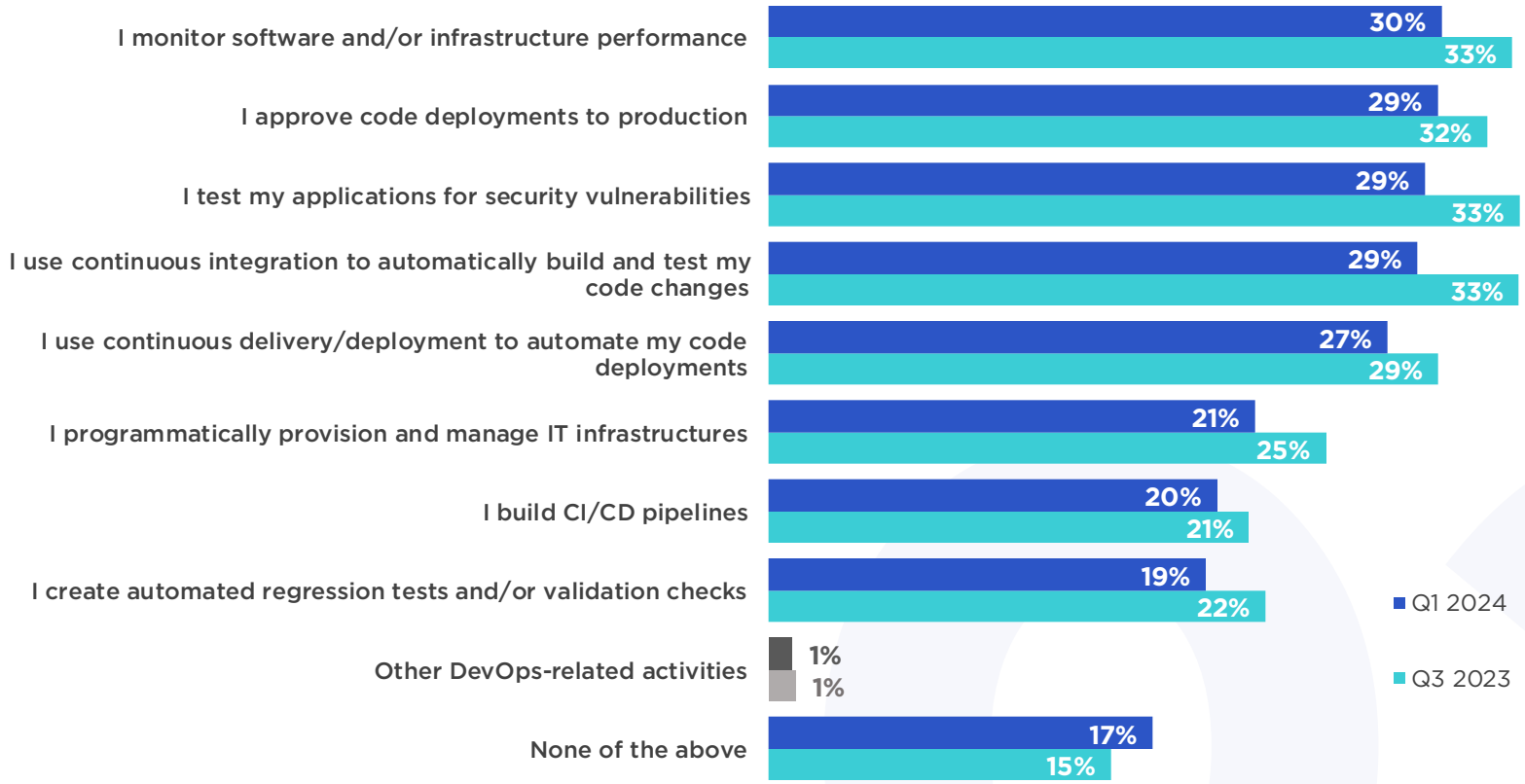
2. Who is into DevOps?

While DevOps-related practices have been widely adopted across all development areas, we see differing involvement rates between sectors. A massive 90% of industrial Internet of Things (IoT) developers are engaged in DevOps-related activities, compared to 80% of games developers. Compared to previous results, the proportion involved in DevOps among machine learning/artificial intelligence (ML/AI) developers has overtaken developers in data science. DevOps involvement has remained stable at 85%, as many new developers in ML/AI are experienced in software development in general. Data science, on the other hand, has dropped from 88% in Q1 2023 to 83% in Q3 2024, as less experienced developers join the space with low rates of DevOps usage.

In terms of organization sizes, our data reveals that medium-sized businesses have the highest involvement in DevOps practices (90%). However, organizations of other sizes have seen significant growth in the proportion of developers undertaking DevOps activities. Both small businesses and large enterprises now have 88% of their developers involved in DevOps-related activities, up by 4 and 3 percentage points, respectively, from Q1 2022. However, the proportion of freelancers involved in DevOps (79%) has been stable for the past two years.

The vast majority (83%) of developers are involved in DevOps practices

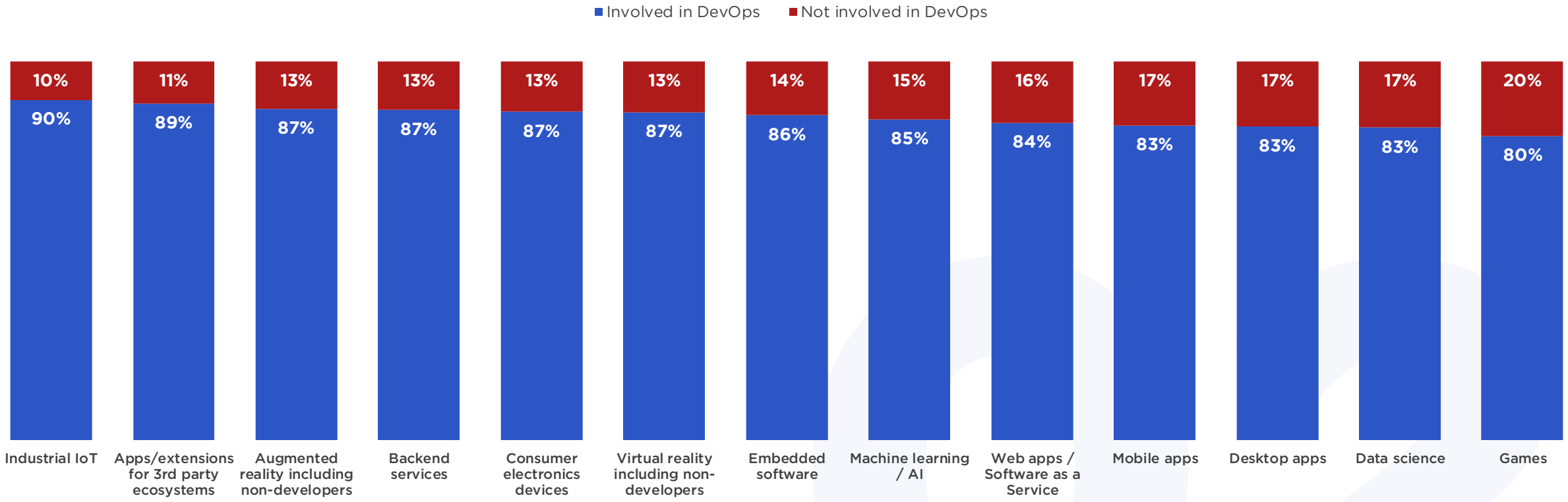
Popularity of DevOps practices



Question wording: Which of the following development activities are you involved in?
% of developers Q3 2023 (n=16,111) | Q1 2024 (n=9,807)

Data science and game developers are the least likely to be involved in DevOps

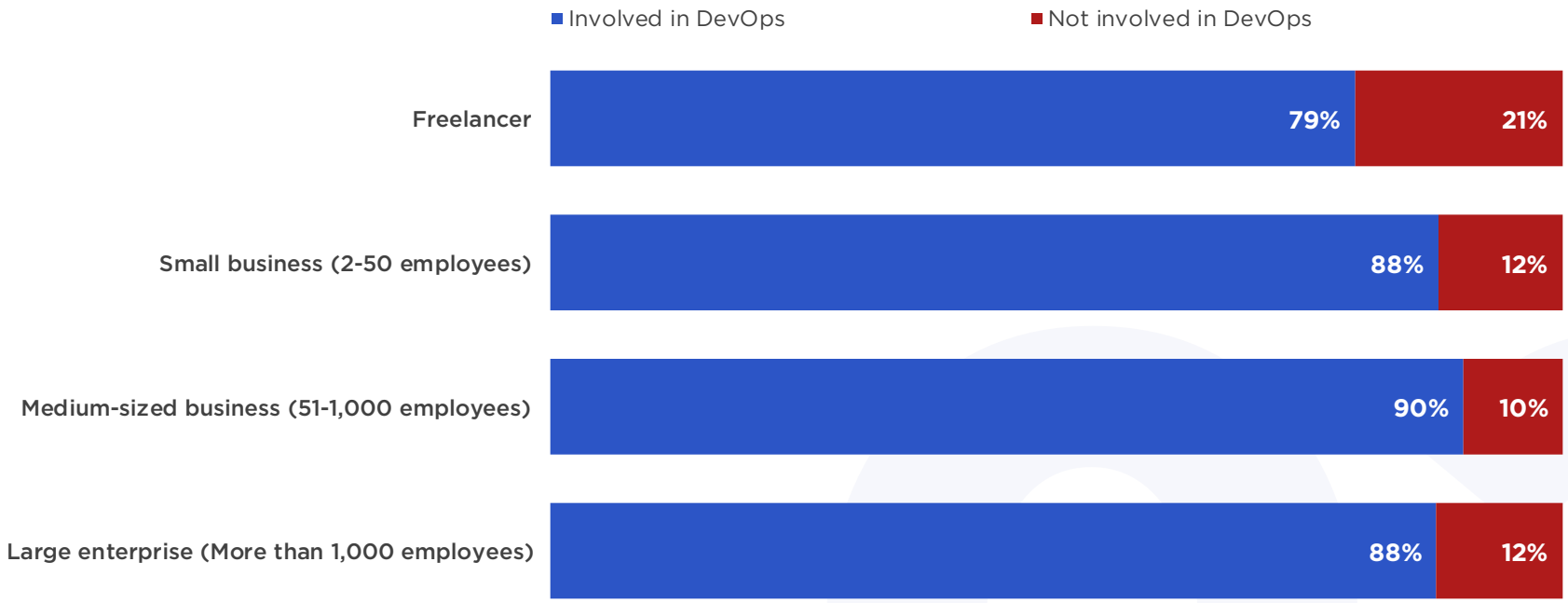
Involvement in DevOps by software development area



Question wording: Which of the following development activities are you involved in?
% of developers involved in DevOps activities by software sector (n=9,807)

Developers across organization sizes show similar levels of involvement in DevOps activities

Involvement in DevOps by software development area



Question wording: Which of the following development activities are you involved in?
% of professional developers involved in DevOps activities by company size (n=7,062)

HOW HAS SOFTWARE DELIVERY PERFORMANCE EVOLVED?

03

3. How Has Software Delivery Performance Evolved?

For individuals and organizations to measure the effectiveness of CD efforts, a set of robust performance metrics is required. In our survey, we ask developers about their performance for three of the four DevOps Research and Assessment (DORA) metrics[1]: lead time for changes, deployment frequency, and time to restore service. These metrics measure software delivery performance and are predictive of organizational performance.

Within the general developer population, our data shows no clear signs that the velocity for code changes has improved over the last three and a half years. Over this period, the percentage of top performers — those with lead times of less than one day — has fluctuated between 13% and 17% and currently stands at 14% in Q1 2024. It is possible that the increase in DevOps practices has not yet trickled down to positively impact performance. However, DevOps has been a mainstay of professional development for several years now. Instead, it may be that the ubiquity of DevOps practices has allowed developers and organizations to increase the complexity of projects they are involved in, counteracting the benefits to development velocity. In other words, DevOps practices have likely made the development velocity of complex projects comparable to simpler projects without DevOps practices.

[1] *Forsgren, N., Humble, J., Kim, G. 2018. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution Press*

3. How Has Software Delivery Performance Evolved?

For deployment frequency, we have seen a continual decrease in the proportion of developers who are top performers — those with multiple deployments per day. However, this decrease has been small over time, which, while concerning, alone does not indicate a huge crisis. The growing proportion of developers who deploy less frequently than once per month is concerning though.

We see a similar trend for the time to restore service after unexpected downtime. The proportion of developers who can restore service in less than an hour has remained at around 11% since Q3 2022, though down from 17% in Q3 2020. However, the proportion of the worst-performing developers – more than one week to restore service – has been steadily increasing and is now the condition for 41% of developers.

While DevOps may be allowing for the scoping of increasingly complex projects that can harm performance here, another factor may be the consolidation of DevOps technology usage. As we will explore in the next section, developers have been steadily reducing the number of different DevOps technologies they have been using. As DevOps matures, developers go from exploring the space to focusing only on the technologies they find most useful. However, usefulness does not always directly correlate to deployment performance.

3. How Has Software Delivery Performance Evolved?

Additional DevOps tools may have made their deployment performance better, but have added additional mental load to their work. A particular example can be found with the well-reported issue of alert fatigue[2]. Using a broad range of tools may have helped developers improve their deployment performance, but came with additional aspects that were not enjoyed, and, as such, are considered less useful to a developer's workflow, even if they are crucial to secure and fast deployments.

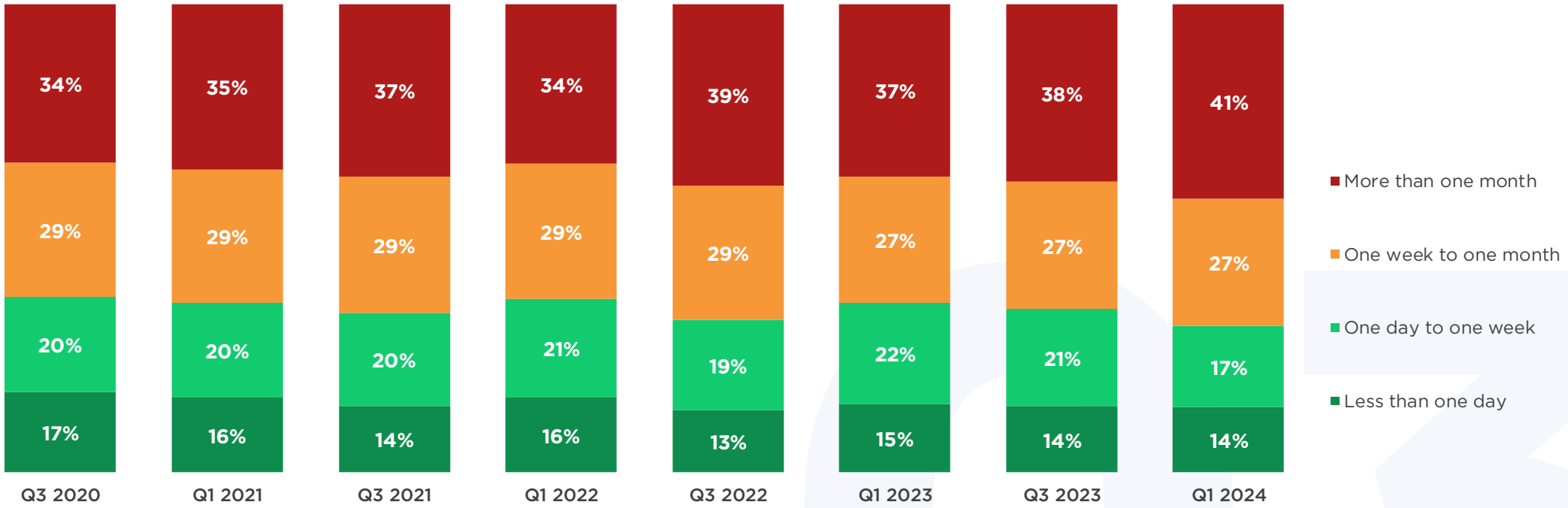
The number of technologies a developer is using in their workflow is strongly correlated to their developer velocity. Particularly, the lowest velocity groups are found in much larger proportions among those using fewer technologies.

The speed and stability metrics remain strongly correlated, rather than one compromising the other. The majority of developers (61%) who are low performers on lead time for code changes are also low performers for service restoration time. At the other end of the spectrum, 37% of those who are top performers for lead time are also top performers for service restoration. A further 44% of those in the second-best service restoration time performance group have a lead time of less than one week, indicating good speed, even if these developers are not top performers on stability.

[2] *Understanding and fighting alert fatigue, Atlassian*

The proportion of low performers for lead time for code changes continues to increase

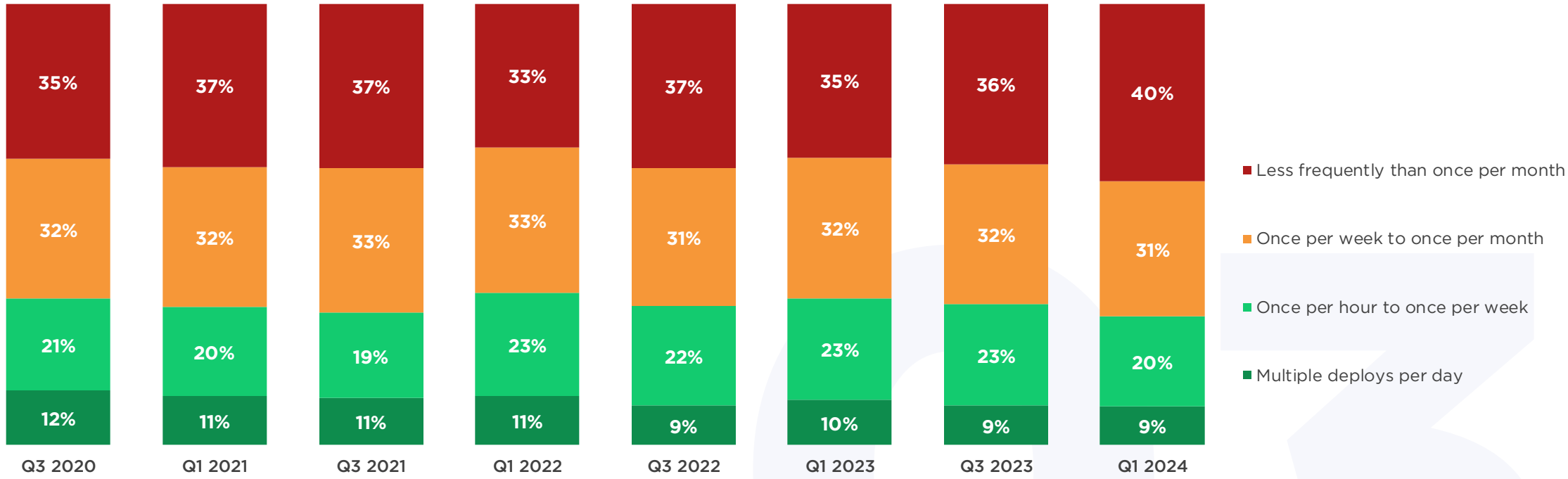
Software delivery performance – Lead time for code changes



Question wording: On average, how long does it take you to go from code committed to code successfully running in production?
% of DevOps practitioners (Q3 2020 n=10,252 | Q1 2021 n=7,814 | Q3 2021 n=8,784 | Q1 2022 n=9,640 | Q3 2022 n=13,108 | Q1 2023 n=13,048 | Q3 2023 n=12,055 | Q1 2024 n=7,276)

The proportion of top performers (9%) has remained stable since Q3 2022

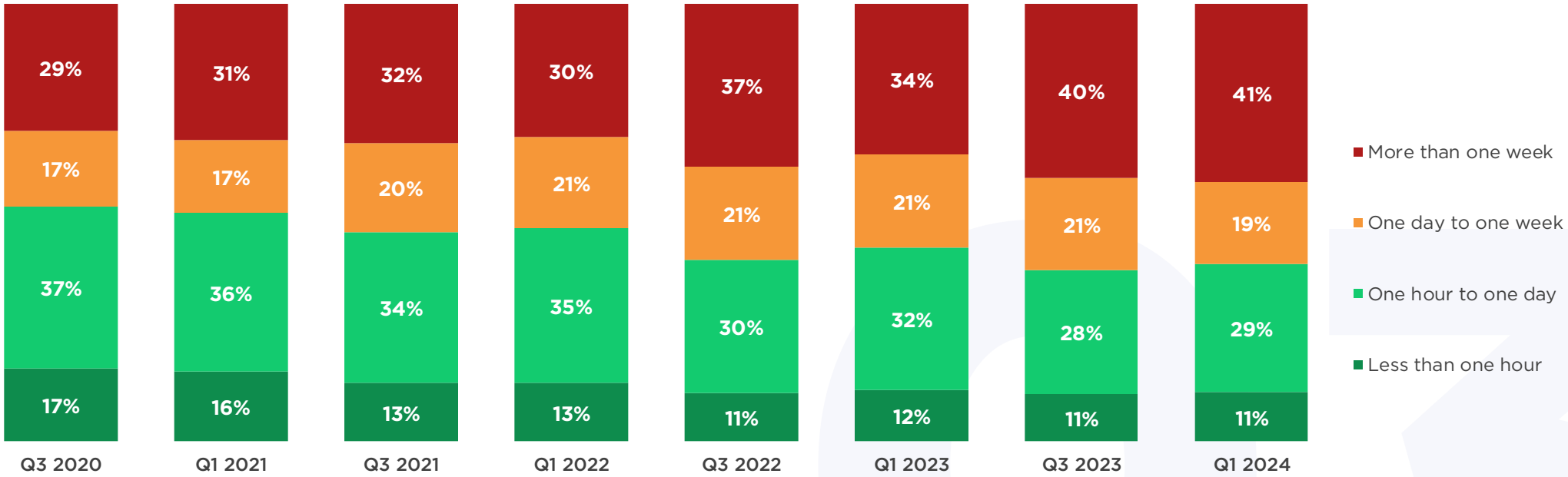
Software delivery performance – Deployment frequency



Question wording: On average, how often do you or your team deploy code to production?
% of DevOps practitioners (Q3 2020 n=10,119 | Q1 2021 n=7,613 | Q3 2021 n=8,619 | Q1 2022 n=9,473 | Q3 2022 n=12,912 | Q1 2023 n=12,747 | Q3 2023 n=11,822 | Q1 2024 n=7,149)

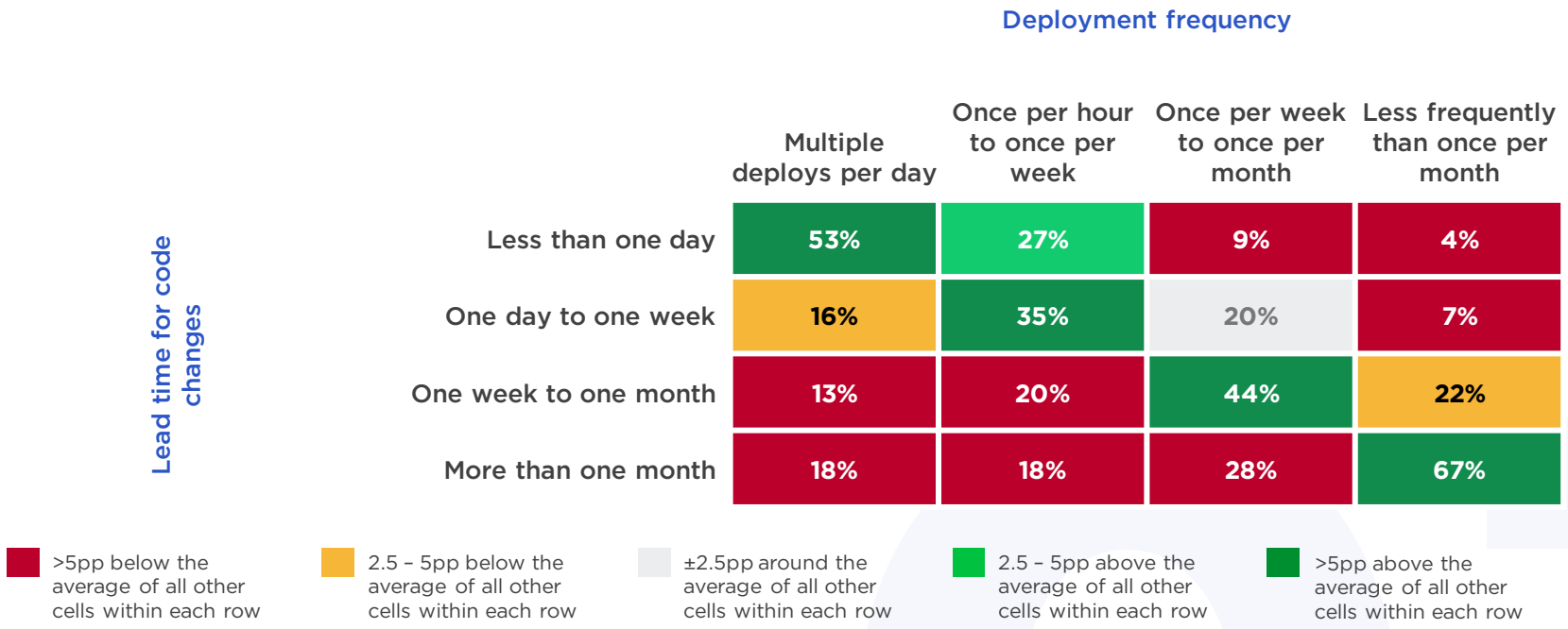
While the proportion of top performers for service restoration has remained stable since Q3 2022, the proportion of low performers has been growing, since Q3 2022

Software delivery performance – Time to restore service



Question wording: On average, how long does it take you or your team to restore service from an unplanned outage or service impairment?
% of DevOps practitioners (Q3 2020 n=9,349 | Q1 2021 n=7,221 | Q3 2021 n=8,126 | Q1 2022 n=8,927 | Q3 2022 n=12,385 | Q1 2023 n=12,250 | Q3 2023 n=11,562 | Q1 2024 n=6,909)

Lead time for code changes and deployment frequency performance are tightly linked



Question wording: On average, how long does it take you to go from code committed to code successfully running in production? | On average, how long does it take you or your team to restore service from an unplanned outage or service impairment?
% of DevOps practitioners (n=6,662)

REFLECTING ON FACTORS DRIVING SOFTWARE DELIVERY PERFORMANCE

04

4.1 DevOps Technology Usage

In our survey, we capture information on a broad range of DevOps-related technologies that developers use, ranging from tools for managing source code to tools for monitoring application performance. The average number of technologies that DevOps practitioners use, of those listed, has decreased recently, from more than four technologies, on average, before Q1 2023 to 3.4 in Q1 2024.

Similar to the trend in DevOps practices, DevOps technologies are showing a steady decrease in usage over time, but they are retaining their relative popularity. Only agile project management tools and AI-assisted coding tools maintained the proportion of developers using them. However, it is important to note that the developer population is continually growing, as is the number of tools available. As such, small decreases in the proportion of developers using DevOps technologies are not necessarily indicative of declining usage

A further impact may be a change in the way developers and teams approach the 'shift left'[1] development philosophy within DevOps. A mixture of security concerns[2], developers feeling inadequately empowered, and DevOps and DevSecOps teams being key to centralizing or managing processes has left developers themselves with fewer DevOps activities in their mandate.

Embracing DevOps as a philosophy, rather than just a collection of technologies, requires time and capital investment to ensure the processes are working to improve developers' experience and velocity. Further evidence for issues with shift left can be found with developers at companies with more than 1,000 employees. The development velocity across all three measured metrics has remained stable over the last three and a half years, while those at smaller organizations have been performing worse.

[1] *"Why 'shift left' is now a dirty term in some security circles", Ericka Chickowski, Reversing Labs*
[2] *"The State of Software Supply Chain Security Maturity", Liam Dodd, SlashData & Red Hat*

4. Reflecting on Factors Driving Software Delivery Performance

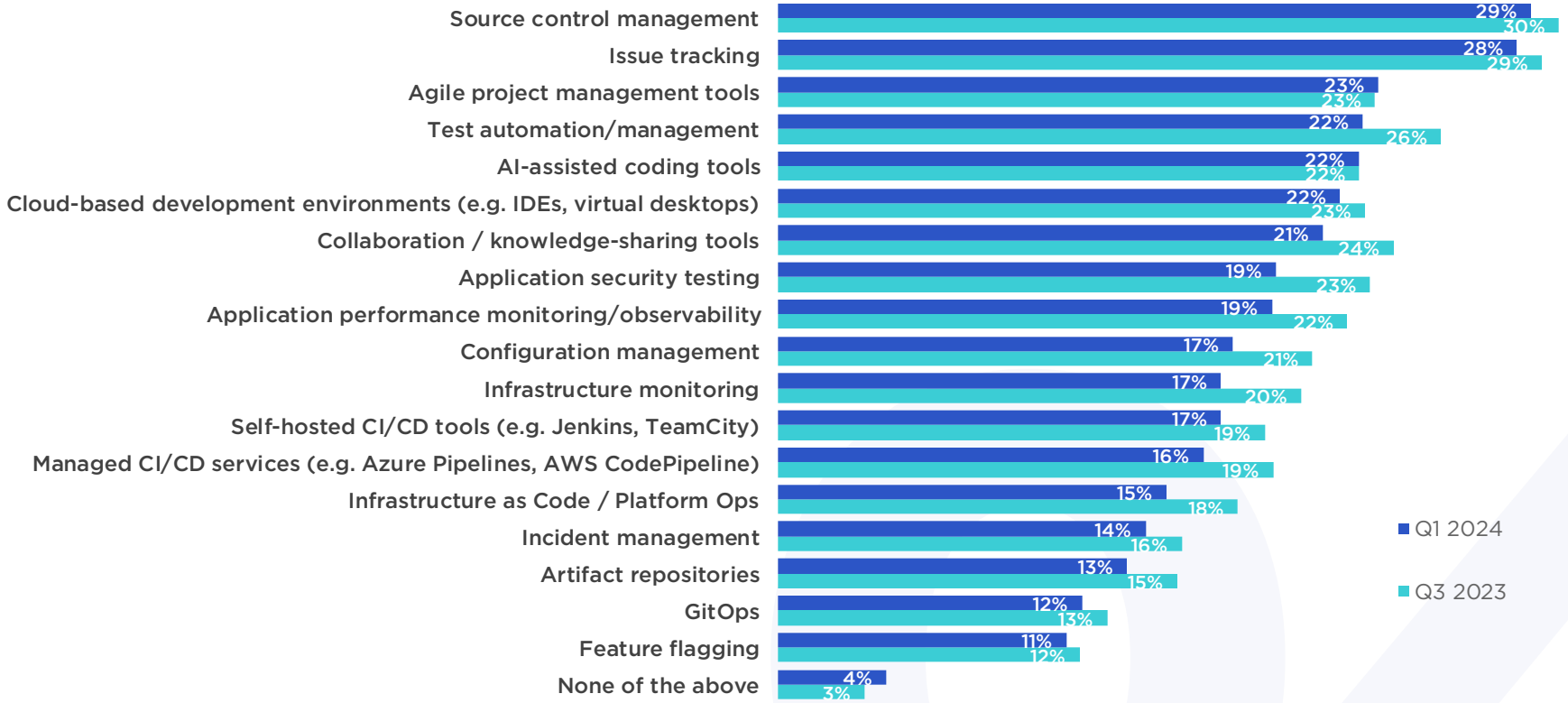
Outside of the relative adoption rates of various DevOps technologies, it is important to understand if the use of more technologies impacts developer performance. Our motivation is to examine whether having developers with a greater breadth of exposure to different aspects of DevOps and technologies to support them helps to collectively drive performance. On the contrary, it may be beneficial for developers to have a narrow and specific focus or responsibility and, therefore, have fewer tools or technologies to manage.

We find that the use of multiple DevOps technologies is strongly correlated with an increase in developer performance. However, while there is an increase in top performers at the larger technology counts, what stands out more is the large decrease in the proportion of low performers.

As of Q1 2024, of those using one technology, 50% are in the low-performance group for lead time for code changes, 47% for deployment frequency, and 57% for time to restore service. For those using ten or more technologies, only 13%, 18%, and 6% are in the low-performing groups for the aforementioned performance metrics, respectively. While a broad usage of DevOps technologies is associated with a greater likelihood of being a top performer, more critically, it reduces the likelihood of being a low performer substantially.

The usage rate of AI-assisted coding and agile project management tools has remained consistent over the last year

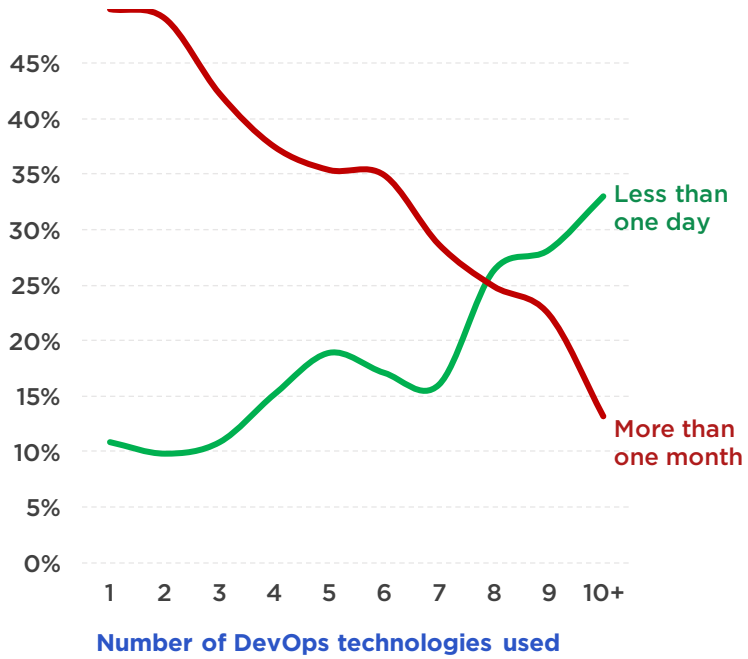
Popularity of DevOps technologies



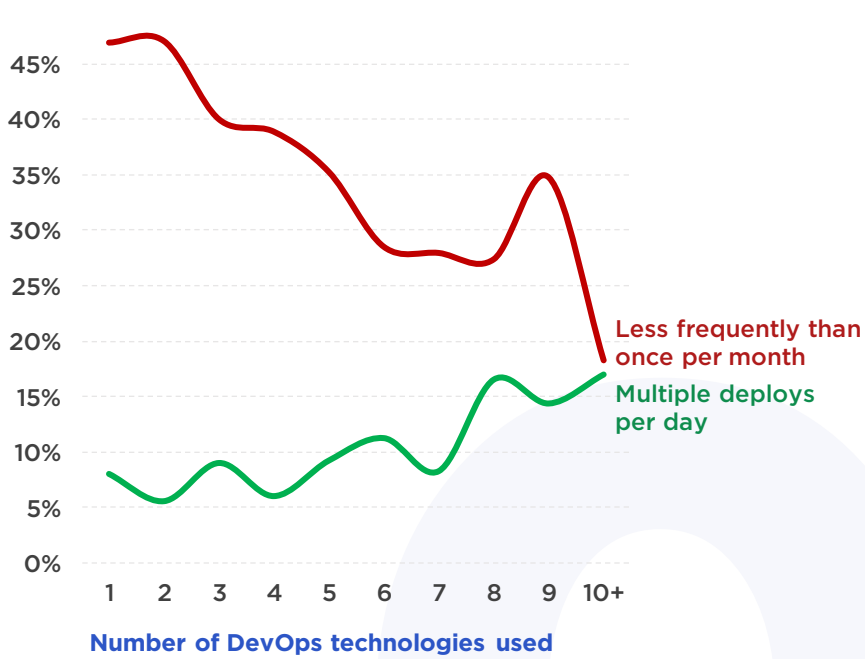
Question wording: Which of the following technologies have you used as part of your development activities in the last 12 months?
% of DevOps practitioners Q3 2023 (n=13,599) | Q1 2024 (n=8,102)

Using a broader range of DevOps technologies is correlated with better delivery performance

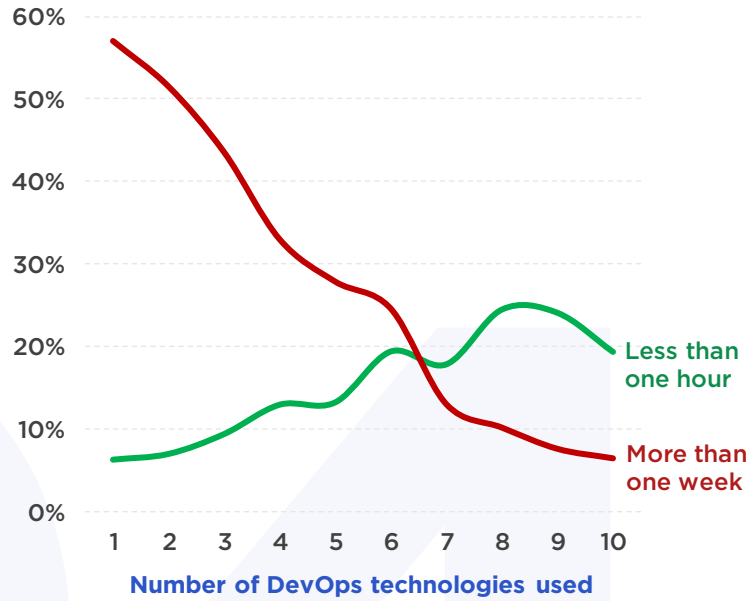
Code lead time for changes against the number of DevOps technologies used



Deployment frequency against the number of DevOps technologies used



Time to restore service against the number of DevOps technologies used



Question wording: Which of the following technologies have you used as part of your development activities in the last 12 months? | On average, how long does it take you to go from code committed to code successfully running in production? | On average, how often do you or your team deploy code to production? | On average, how long does it take you or your team to restore service from an unplanned outage or service impairment?
% of DevOps practitioners by number of DevOps technologies used (n=6,930)

4.2 Usage and the Interoperability of Self-Hosted and Managed Platforms

While the breadth of technology usage has been shown to benefit development velocity, certain tools are found to be more impactful than others. When comparing DevOps practitioners who use CI/CD tools — managed or self-hosted against those who do not – we can see a distinct difference in performance across all three development metrics.

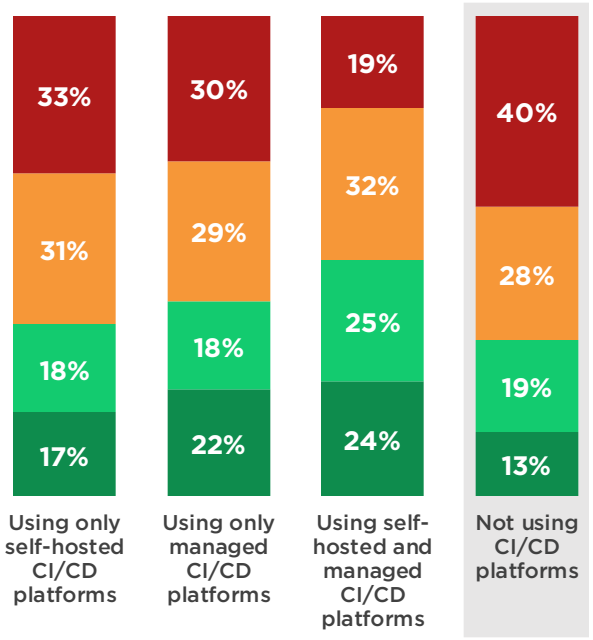
Developers who use CI/CD tools are more likely to be top performers compared to those who do not. This is particularly stark for top performers in time to restore service, where those who use CI/CD tools are substantially more likely (a minimum of 15%) to be top performers than those who do not (11%). Even more distinctly, those not using any CI/CD tools are more likely to be low performers than those who use any CI/CD tool. This is most clear on the lead-time for code change, where 40% of those not using any CI/CD tools are low performers, compared to the next largest CI/CD configuration, 33% of those using only self-hosted CI/CD platforms.

Further, we divide developers based on whether they are using self-hosted or managed CI/CD platforms, or if they are using both. For all three DORA measures, those using managed CI/CD platforms are less likely to be low performers than those using only self-hosted solutions (30% vs. 33% for lead time, 34% vs. 32% for deployment frequency, and 29% vs. 25% for service restoration). Managed CI/CD platforms may offer both a scalability benefit that requires less manual intervention than a self-hosted platform, as well as allow developers to focus more on their core competencies rather than managing infrastructure.

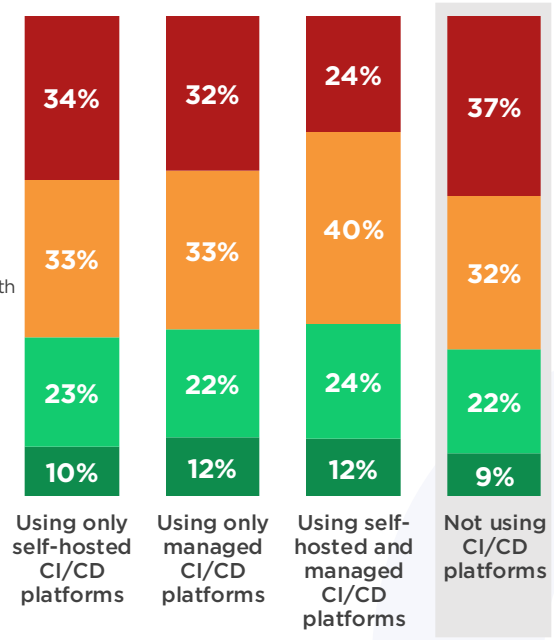
However, those using both self-hosted and managed CI/CD platforms perform the best across all DORA metrics. These developers may be able to leverage the best of both approaches, combining aspects like the convenience and built-in features of a managed platform with customizations in their self-hosted CI/CD platforms as and where needed for greater efficiencies. These developers may also be positioned to make greater optimizations to their CI/CD practices. Selecting when to use their managed or self-hosted platform may induce more natural learning moments, which helps develop an iterative learning process that generates greater improvements in their processes.

Using both self-hosted and managed CI/CD platforms is associated with better delivery performance

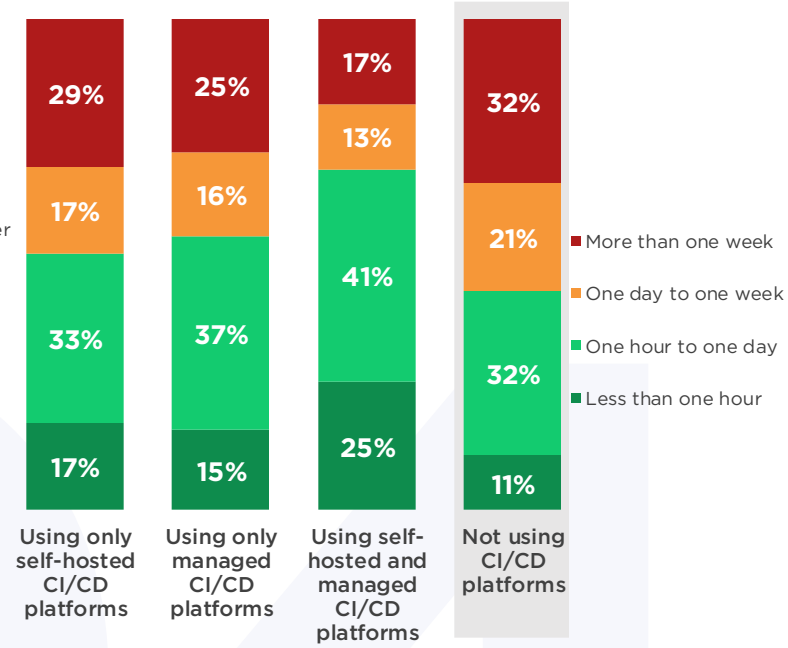
Code lead time for changes performance



Deployment frequency performance



Time to restore service performance



Question wording: Which of the following technologies have you used as part of your development activities in the last 12 months? | On average, how long does it take you to go from code committed to code successfully running in production? | On average, how often do you or your team deploy code to production? | On average, how long does it take you or your team to restore service from an unplanned outage or service impairment?
% of DevOps practitioners for each CI/CD configuration (n=6,930)

4.3 Multiple CI/CD Platform Interoperability

We have shown that using CI/CD tools increases the likelihood of DevOps practitioners being top performers across lead time for code changes, deployment frequency, and time to restore service. While we have seen that using managed and self-hosted platforms in conjunction is beneficial, there are concerns that using multiple CI/CD platforms of the same form may introduce interoperability issues.

Across all three of the delivery performance metrics used, we can see that an increased number of self-hosted CI/CD tools used is not associated with greater performance. This trend is similar for using multiple managed CI/CD platforms, though less extreme for some delivery performance metrics.

For lead time for code changes and deployment, we see that increasing the number of self-hosted CI/CD platforms used does not improve performance. In particular, we see a drop in top performers, down to 10%, for those using three self-hosted CI/CD platforms in their workflow. We see similar, but less severe, troughs for developers using three CI/CD platforms across deployment frequency and service restoration time. On the other hand, the proportion of low performers increases dramatically with an increasing number of self-hosted CI/CD tools used. This suggests that there is a diminishing return from increasing the number of CI/CD tools a developer uses.

4. Reflecting on Factors Driving Software Delivery Performance

For managed CI/CD tools, the introduction of additional tools leads to a decrease in the proportion of top performers on lead time for code change performance. While the proportion of low performers increases for more than one managed CI/CD platform, the proportion remains consistent from two to four managed CI/CD platforms. This suggests that the complexity of interoperability issues emerges from the introduction of a single additional platform, but further additions don't increase the issues at the same rate for low lead-time performance.

The usage of an increasing number of tools may also be a response to increased complexity, which is having negative impacts on the performance of these developers. Similarly, the integration of multiple tools may not be optimally implemented, especially when using self-hosted tools, leading to function overlap that is impacting performance.

On developers' deployment frequency, the proportion of low performers increases as more managed CI/CD platforms are introduced to the workflow simultaneously. However, the proportion of top performers overall is greater for those using more than five platforms (13%), compared to those just using one (11%). However, the increase is very small and suggests that using multiple managed CI/CD platforms has minimal impact on improving deployment frequency, but is much more likely to lead to an increase in low performance.

The time to restore service metric sees the most dramatic increase in low performers from the increased number of self-hosted CI/CD tools used. Among those using one self-hosted CI/CD tool, the likelihood of being a top performer (25%) is greater than being a low performer (18%). However, once a practitioner passes two or three self-hosted CI/CD tools, this reverses dramatically. Among developers who use more than five tools, 62% are low performers, while only 14% are top performers.

4. Reflecting on Factors Driving Software Delivery Performance

An increasing number of self-hosted tools used having such a strongly negative impact on service restoration time has multiple possible explanations. However, interoperability issues may be at the center of many of them. Multiple tools may make it challenging to integrate all of them well, leading to a greater challenge to isolate the service-impacting issue at hand.

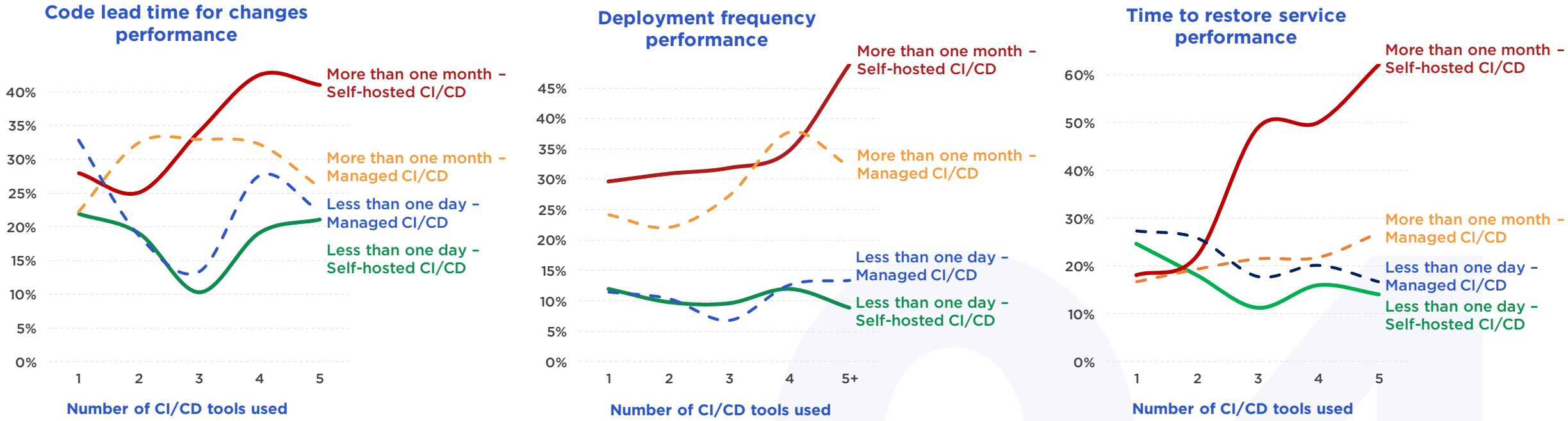
While the time to restore service also sees a negative impact of an increased number of managed CI/CD platforms in their workflow, it is less severe than the other DORA metrics, especially when contrasted with lead time for code changes. Further, compared to the impact of multiple self-hosted tools, the negative impact is much smaller. One possible reason for this may be that lead time for code changes is impacted not just by platform usage, but also by their CI and development processes. Multiple managed CI/CD platforms may introduce fragmentation of the CI process, leading to greater negative impacts. Similarly, development practices like code review, collaboration, and testing may be impacted by having to adapt to multiple platforms throughout the workflow, and this challenge has a larger impact on lead-time performance.

On the other hand, even though multiple managed CI/CD platforms add complexity to the service restoration process, the features commonly associated with managed CI/CD platforms may mitigate the negative impact. For example, managed CI/CD platforms often have automated rollback and forward mechanisms or isolated production environments that would improve the time to restore service. Multiple CI/CD platforms create interoperability issues for isolating the cause of the issue and restoring service correctly, but the platforms themselves positively impact service restoration.

Compared to self-hosted tools, a lack of standardization between tools may make it more difficult for all tools to work together well, which also increases the challenge of addressing service failure. Providing a much larger issue from interoperability than managed tools.

The use of multiple self-hosted CI/CD tools can lead to longer response times to service outages

Impact of the number of self-hosted CI/CD platforms used on delivery performance



Question wording: Which of the following self-hosted CI/CD tools have you used in the past 12 months? | Which of the following managed CI/CD services are you aware of or currently using? | On average, how long does it take you to go from code committed to code successfully running in production? | On average, how often do you or your team deploy code to production? | On average, how long does it take you or your team to restore service from an unplanned outage or service impairment?
% of DevOps practitioners for each performance metric using multiple CI/CD tools simultaneously (Self-hosted: n=1.295, Managed: n=1,192)

4.4 Developer Experience

As highlighted earlier, developers with less experience in software development are not adopting DevOps practices and technologies at rates comparable to their more experienced colleagues. While it may be expected that more experienced developers would add more DevOps activities to their workflow as they work on more complex projects or gain increased levels of ownership or responsibility, the substantially lower adoption among newer developers is concerning.

Modern software development benefits greatly from developers and organizations engaging with the DevOps philosophy and the technologies associated with it. Developers adopting these early in their career allows them to benefit from them immediately, as well as customizing their development workflow as their career progresses, without having to take large learning leaps as they aim to grow.

However, looking at the DORA metrics for developers based on experience may indicate organizations and teams failing to build development teams where less experienced developers are able to benefit from more experienced team members.

More experienced developers are more likely to be top performers across all delivery performance metrics measured. On lead time for code changes, developers with 11 to 15 (22%) and more than 16 (21%) years of experience are around twice as likely to be top performers compared to their less experienced counterparts, approximately 10% of those with five or fewer years of experience.

4. Reflecting on Factors Driving Software Delivery Performance

For time to restore service, the effect is even larger, with only 5% of those with two or fewer years of experience being top performers, compared to 16% of those with 11 to 15 and 22% of those with 16 or more years of experience. On the other hand, deployment frequency top performers are still a larger proportion of the more experienced developers, but the impact of experience is not as large as on the other two performance metrics.

However, while the increase in top performers as developers gain more experience is substantial, the decrease in low performers is much larger. On lead time for code changes, low performers go from 56% of the least experienced developers to 29% of the most experienced. We see the same trend for deployment frequency, 51% to 34%, and most extremely for service restoration, from 67% to just 9%.

While it is expected that more experienced developers will generally be better performers, particularly with the nature of building software for release within organizations, this is deeply concerning due to DORA metrics focusing on the team-scale capabilities. This suggests two possible concerning situations in the development space: a lack of team or organization cohesion on processes, or less experienced developers being unaware of their organization's processes.

Less experienced developers may, overall, be less aware of the processes that their team use, and as such are unaware of the velocity at which their team can deliver. This can lead to worse performance, as they are not using the best methods available to them, or are working in isolation of the DevOps practices their organization uses. This can also be impacted by the centralization of DevOps to specific teams or team members, meaning that these less experienced developers are unaware of, for example, how often their team deploys code changes because they themselves deploy infrequently.

4. Reflecting on Factors Driving Software Delivery Performance

This would be made worse by the second factor, the lack of cohesion from an organization level. As highlighted in other research[1], while developers may often strive to engage in the best practices, they are let down by their organizations not enforcing or standardizing best practices throughout the organization. Less experienced developers are unlikely to be working alone, without either supervision or collaboration with more experienced developers. If more experienced developers can have better delivery performance metric results, it suggests that effective methods are not being shared with newer developers.

More experienced developers are using a larger number of DevOps technologies on average than those with less experience. Developers with less than two years of experience are using 2.3 tools on average, compared to those with more than 16 years of experience using an average of 5.2. On top of this, they are adopting technologies at much higher rates. For example, 65% of developers with more than 16 years of experience are using source control management tools, compared to just 22% of those with between three and five years of experience, and even less of those with fewer years of experience.

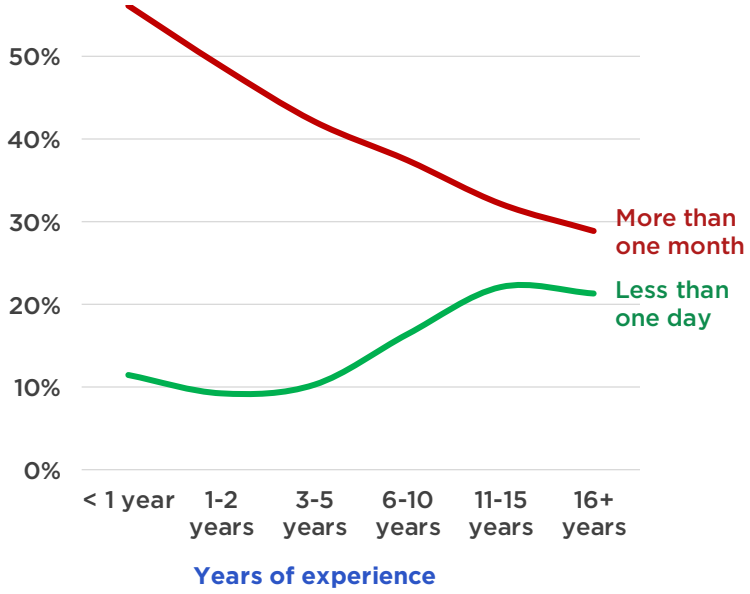
Without organizations determining standardized approaches that all teams are aware of and involved in, they risk having their ability to utilize the full benefits of both DevOps and CI/CD approaches undermined. Experienced developers are able to leverage technologies to their benefit while less experienced developers are left behind.

[2] "The State of Software Supply Chain Security Maturity", Liam Dodd, SlashData & Red Hat

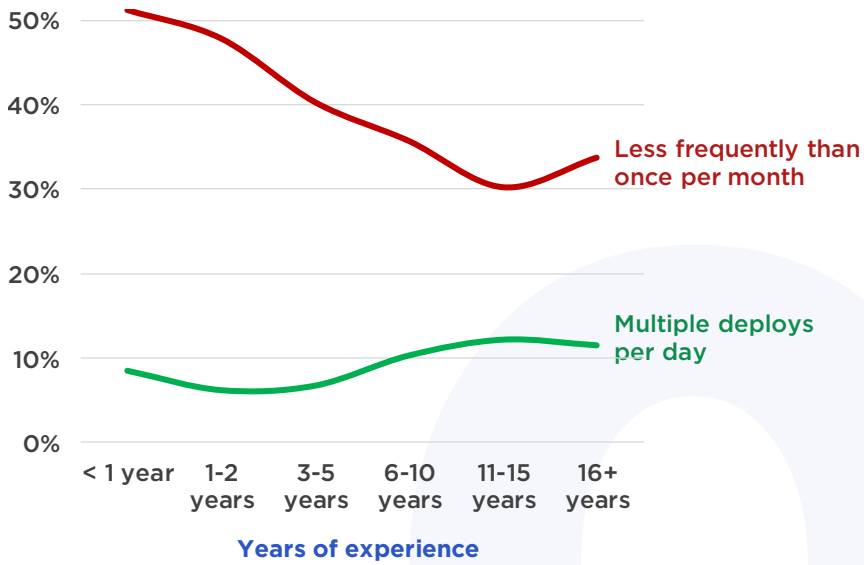
The likelihood of being a poor performer decreases significantly as developers' experience in software development increases

Impact of the number of self-hosted CI/CD platforms used on delivery performance

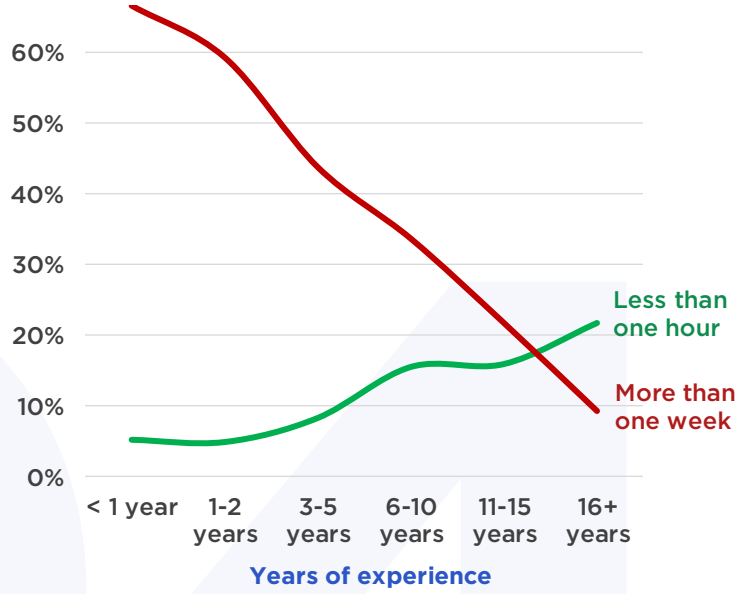
Code lead time for changes performance



Deployment frequency performance



Time to restore service performance



Question wording: How many years have you been working in software development in general? | On average, how long does it take you to go from code committed to code successfully running in production? | On average, how often do you or your team deploy code to production? | On average, how long does it take you or your team to restore service from an unplanned outage or service impairment?
% of DevOps practitioners for each performance metric using multiple CI/CD tools simultaneously (Self-hosted: n=1.295, Managed: n=1,272)

CONCLUSION

05

5. Conclusion

This report adds to the existing strong base of evidence that shows embracing DevOps can help developers and organizations improve their software delivery performance. Using a broad range of DevOps technologies is associated with better performance, as is the specific use of CI/CD tools. In particular, the use of managed and self-hosted CI/CD tools together is associated with the best delivery performance results.

However, we see a trend of decreasing involvement in DevOps by developers, as well as fewer DevOps technologies being used. This is most prevalent among the newest developers, which, if not addressed, can propagate a cohort of developers with less exposure and comfort with DevOps, which can both impact their personal development and organization performance.

A key factor underlying the observations in this report is the important role that organizations and team leaders play in guiding their teams to greater performance. Whether it is helping developers consolidate their workflows down to a single self-hosted and/or managed CI/CD tool, to prevent interoperability issues, or ensuring that newer developers are more familiar with DevOps in general as well as the practices used in the organization. This also opens a position for organizations to implement and build best practices throughout their organization to ensure that delivery performance is not dependent on a developer's individual experience, and instead one that can leverage the talent of the most experienced developers to elevate the less experienced.

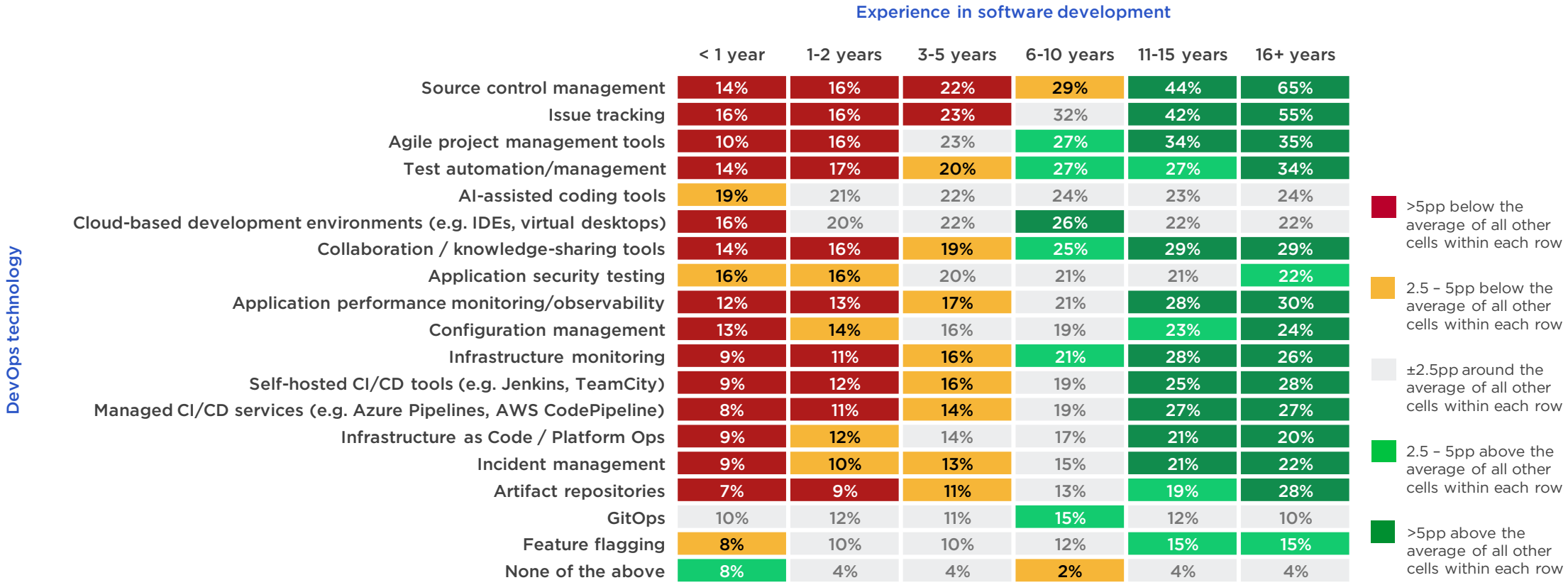
5. Conclusion

Finally, the decrease in the number of technologies developers are using, despite their benefit to delivery performance, may emerge from developers themselves being less likely to be motivated by delivery performance than by their personal comfort with a workflow. Vendors providing DevOps tools need to keep developer fatigue front of mind when considering how technologies are used to prevent developers from turning away from them. Similarly, as the use of a broad range of DevOps technologies shows a large impact on developer performance, vendors should ensure their tools can be integrated with other technologies seamlessly, and advertise this feature too.

Organizations and team leaders should also place developer experience at the center of their plans for improving delivery performance. The benefits have been demonstrated, but they are unlikely to amount to much if developers ignore or sidestep these processes due to them negatively impacting their experience developing. Balancing what processes to shift-left and which to maintain as the responsibility of specific teams and developers is likely a more effective strategy for achieving improved developer and delivery performance.

APPENDIX

More experienced developers are more likely to be using more DevOps technologies and at a greater rate of adoption



Question wording: How many years have you been working in software development in general? | Which of the following technologies have you used as part of your development activities in the last 12 months?
% of DevOps practitioners for each experience group using DevOps technologies (n=8,102)

METHODOLOGY

The Developer Nation Survey

The 26th edition of the Developer Nation survey reached more than 10,000 respondents from 135 countries around the world. As such, the Developer Nation series of surveys continues to be the most global independent research on mobile, desktop, industrial IoT, consumer electronics, embedded, third-party app ecosystems, cloud, web, game, augmented and virtual reality, and machine learning developers and data scientists combined, ever conducted. The report is based on a large-scale, online developer survey designed, produced, and carried out by SlashData over a period of ten weeks between November 2023 and February 2024.

Respondents to the online survey came from 136 countries, including major app and machine learning development hotspots such as the US, China, India, Israel, and the UK, even stretching all the way to Kenya, Brazil, and Jordan. The geographic reach of this survey is truly reflective of the global scale of the developer economy. The online survey was translated into nine languages in addition to English, namely simplified Chinese, traditional Chinese, French, Spanish, Portuguese, Vietnamese, Russian, Japanese, and Korean, and was promoted by more than 75 leading community and media partners within the software development industry.

Our respondents came from a broad age spectrum, from young coders and creators who are under 18 to the seasoned ones over 55.

Respondents were asked which types of projects they are involved in out of the 13 under study, namely web apps / SaaS, mobile apps, desktop apps, backend services, augmented reality, virtual reality, games, data science, machine learning / artificial intelligence, industrial IoT, consumer electronics devices, embedded software, and apps/extensions for third-party app ecosystems. They also told us if they are into their areas of involvement as professionals, hobbyists, or students - or as any combination of these - and how many years of experience they have in each.

To eliminate the effect of regional sampling biases, we first weighted to correct for over-represented individual countries within regions. We then weighted the regional distribution across nine regions by a factor that was determined by the regional distribution and growth trends identified in our Developer Nation research. Each of the separate branches: mobile, desktop, Industrial IoT, consumer electronics, embedded software, third-party app ecosystems, cloud, web, games, augmented and virtual reality, and data science and machine learning were weighted independently and then combined.

To minimise other important sampling biases across our outreach channels, we weighted the responses to derive a representative distribution for technologies used and developer segments. Using ensemble modelling methods, we derived a weighted distribution based on data from independent, representative channels, excluding the channels of our research partners, to eliminate sampling bias due to respondents who were recruited via these channels. Again, this was performed separately for each of mobile, industrial IoT, consumer electronics, embedded software, third-party app ecosystems, desktop, cloud, web, games, augmented and virtual reality, and data science and machine learning.

For more information on our methodology please visit
Our methodology page

/DATA

We help the world understand developers

We survey 30,000+ developers annually – across Web, Desktop, Cloud, Mobile, Industrial IoT, AR/VR, Machine Learning and Data Science, Games, Consumer Electronics and Apps/Extensions for 3rd party ecosystems - to help companies understand who developers are, what they buy and where they are going next.

WHO DEVELOPERS ARE

Developer population sizing
Developer segmentation

WHAT THEY BUY

Why developers are adopting competitor products – and how you can fix that

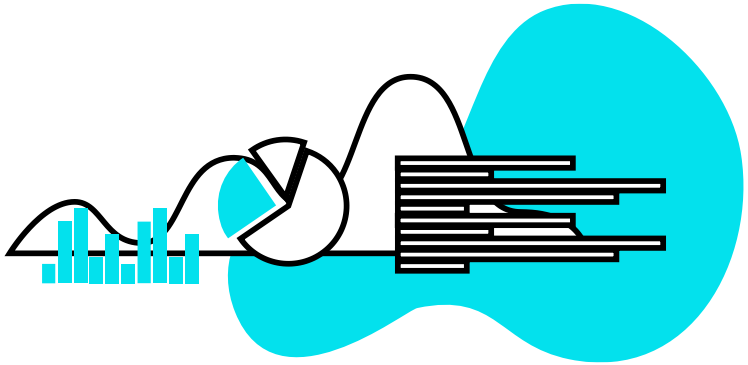
WHERE THEY ARE GOING

Emerging platforms – augmented & virtual reality, machine learning

TRUSTED BY

the leading tech platforms

OUR CLIENTS



/DATA

THE ANALYST OF THE DEVELOPER ECONOMY

www.slashdata.co