

DEVELOPER ECONOMICS

STATE OF THE DEVELOPER NATION 19TH EDITION

The latest trends from our Q3 2020
survey of 17,000+ developers



/DATA

19TH EDITION
PUBLISHED
OCTOBER 2020


About SlashData

SlashData is the leading analyst company in the developer economy, tracking global software developer trends based on more than 30,000 software developers annually in over 160 countries. Our surveys track the changing landscape of mobile, IoT, desktop, cloud, web, AR, VR, games, machine learning developers, and data scientists. Our mantra:

**We help the world understand developers -
and developers understand the world.**

SlashData Ltd.
19-21 Hatton Gardens,
London EC1N 8BA
United Kingdom
+44 845 003 8742

www.slashdata.co

 @SlashDataHQ

About Developer Economics

Developer Economics is a global community engaging thousands of developers of all shapes and sizes across the globe, enabling them to benchmark themselves against the developer nation. We are committed to facilitating community contribution and knowledge sharing, and promoting diversity and inclusion in the developer ecosystem.

Our vision is to empower developers to shape the future!

www.developereconomics.com

 [@DevEconomics](https://twitter.com/DevEconomics)

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 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.

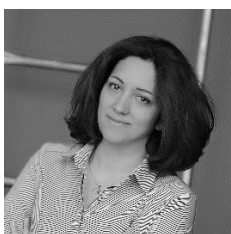
NoDerivatives — you can not remix or transform the content of the report. You may not distribute modified content.

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.

About the authors



Christina Voskoglou

Senior Director of Research

Christina is responsible for all SlashData's research products and heads the analyst and operations teams. With more than 18 years of experience in data mining, BI and CRM design, she leads research planning and methodology, survey design, data analysis, insights generation, and research commercialisation. Christina is also behind SlashData's outcome-based developer segmentation model and is the leading SlashData researcher in machine learning and data science.

christina@slashdata.co | [@ChristinaVoskog](https://twitter.com/ChristinaVoskog)



Jed Stephens

Data Storyteller

Jed has several years of research experience in the academic and industrial sectors mainly focussing on applied statistical research and computational implementations of new statistical methods. He holds an MSc in Statistics. His interest is in turning data into informed actionable decisions.

jed.stephens@slashdata.co



Konstantinos Korakitis

Research Operations Manager

Konstantinos is responsible for driving actionable insights on global developer trends out of a wealth of data points from SlashData's large-scale developer surveys. His analytical approach helps businesses make strategic decisions based on trustworthy developer data. Konstantinos holds a Master's degree in electrical engineering and has nearly 10 years of experience as an engineer, project manager, and data scientist.

konstantinos@slashdata.co



Richard Muir

Data Journalist

Richard is a Data Journalist with over five years' experience working in a range of industries, from government policy to automotive. He is responsible for finding and interpreting actionable insights from SlashData's developer surveys.

richard@slashdata.co | [@rmuir_data](https://twitter.com/rmuir_data)



Sara Iacozza

Data Storyteller

Sara holds a PhD in Cognitive Neuroscience and has more than five years of experience in data analysis and research design. As such, she is an expert in developing research strategies that are both creative and goal oriented. Combining communication skills with her passion for data analytics allows her to clearly convey substantial and actionable insights from any dataset presented to her. In her spare time, Sara promotes gender diversity in the data science community as the cofounder & co-organiser of R-Ladies Nijmegen and R-Ladies Bari.

saraiac@slashdata.co | [@IacozzaSara](https://twitter.com/IacozzaSara)

Table of Contents

Click on the page number to easily navigate to the section.

	About this report	7
	Partners	8
	Key insights	9
01	Developers' extra needs due to COVID-19	10
02	Programming language communities – an update	16
03	Why do developers adopt or reject cloud technologies?	21
04	Who is into DevOps?	27
05	What do developers value in open source?	34
06	Emerging technologies	41
	Methodology	47

About this report

SlashData Developer Economics is the leading research programme on mobile, desktop, industrial IoT, consumer electronics, third party app ecosystems, cloud, web, game, AR/VR and machine learning developers, as well as data scientists, tracking the developer experience across platforms, revenues, apps, languages, tools, APIs, segments, and regions.

The 19th Developer Economics global survey wave ran from June to August 2020 and reached more than 17,000 developers in 159 countries. This research report delves into key developer trends for Q3 2020 and beyond.

The report focuses on **six major themes** - each with its own visualisations - showing how the data lends insight into the developer community.

1. Developers' extra needs due to

COVID-19: Working and performing during a pandemic will leave deep marks behind, both financially and psychologically speaking. In this chapter, we explore COVID-19's effects on developers' changing needs in relation to their development activities.

2. Language communities - an update:

Programming languages are often the kernels of strong communities and the subject of opinionated debate. In this chapter, we provide updated estimates of the number of active software developers using each of the major programming languages, across the globe and across all kinds of programmers.

3. Why do developers adopt or reject cloud technologies?:

In a world where infrastructure can be provisioned and destroyed at will, and where data and server configurations can be transferred easily between homogeneous systems, cloud providers have to find other areas of differentiation in order to compete. Vendor lock-in is much less of an issue for users than it once was, and the rise of the developer as a decision-maker has put even more power into their hands. In this chapter, we look at some of the reasons that developers give for adopting or rejecting different cloud technologies and provide insight into why things are as they are.

4. Who is into DevOps?: DevOps is commonly used as a catch-all term to describe a cultural shift within organisations that enables developers to release software faster and more reliably. However, DevOps is not a single, coherent sector or technology, which often creates confusion as to who is considered a DevOps practitioner. In this chapter, we offer a fresh view on who is into DevOps based on the activities developers are involved in. We also look at the specific roles and software sectors that are most associated with the DevOps culture.

5. What do developers value in open source?:

Based on our research, the use of open-source software (OSS) is ubiquitous in the global developer community. In this chapter, we explore what exactly developers value in using OSS. We also highlight some uncertainties around the future of the open-source movement by presenting trends across geographic regions and software sectors.

6. Emerging technologies: As interest in a technology waxes and wanes, so does its influence. The hot topic of yesterday becomes insignificant in the face of new challenges and opportunities. In our surveys, we have tracked engagement with and adoption of emerging technologies for the past two and a half years. In this chapter, we discuss which technologies have increased and decreased in popularity over the previous twelve months.

We hope you will enjoy this report and find the insights useful! If you have any questions or comments, or are looking for additional data, you can get in touch with Miljana Mitic, Digital Marketing Executive for SlashData, at miljana@slashdata.co. You can download this report for free at

<https://www.developereconomics.com/resources/reports/>

Alex, Anastasia, Andreas, Christina, David, Eitan, Eve, Giannos, Jed, Konstantinos, Lazaros, Maria, Miljana, Moschoula, Natasa, Richard, Sam, Sara, Sarah, Sartios, Sofia, Stathis, Steve M., Steve V., Vanessa, and Viktorija at SlashData.

We'd like to thank everyone who helped us reach 17,000+ respondents for our survey and create this report. Our Media Partners - DigitalOcean, HERE, Huawei, Microsoft, Samsung, and so many others.

A special thanks to the Meetups participating in our survey including: AI Festival Nigeria, BORDERS:NONE, Migracode, R-Ladies Amsterdam, R-Ladies Bari, R-Ladies Milan, R-Ladies Nijmegen, and Social Hackers Academy.



PARTNERS

Our Developer Committee supported our efforts once more, to create the most up-to-date and detailed survey. Special thanks to our members: Amulya, Aydin, Baldomero, Cesar, Christopher, Deborah, Dominic, Hai, Junil, Marcel, Nicholas, Numb, Robert, and Victor, for your help with reviewing survey content, translations, and suggesting prizes.

Our linguistics partner, Palex Group, supported us to create an inclusive survey, translated into eight different languages - Simplified Chinese, Traditional Chinese, Japanese, Korean, Portuguese, Russian, Spanish, and Vietnamese.

Key insights:

Developers' extra needs due to COVID-19

- Four in ten developers report that they need more flexibility in working hours/workload as a consequence of COVID-19.
- Collaboration tools and platforms are the most important technical need for developers.
- Self-employed developers and developers who work for small companies have fewer extra needs due to COVID-19.
- The larger the company developers work for, the stronger their need for self-management and collaboration tools, as well as for mental health support.
- Developers responsible for tooling specifications and for approving budgets and expenses are in the greatest need of increased security, performance, and cloud space.

Who is into DevOps?

- The vast majority of professional developers (more than 80%) are involved in DevOps in one way or another.
- Continuous integration (CI) and continuous deployment (CD) are two of the most common DevOps practices, but only one in four developers use both to fully automate their workflow.
- Programmers are very likely to use CI/CD, but not so much operational practices such as monitoring applications in production environments.
- With some exceptions, software sectors with high concentrations of experienced professionals are more likely to embrace the DevOps model.

Programming language communities – an update

- JavaScript is the most popular programming language by a wide margin, with 12.4M developers globally using it.
- Python now counts 9M users, after adding 2.2M net new developers in the past year alone, outranking Java at the beginning 2020.
- Kotlin is one of the fastest growing language communities, having increased more than two-fold in size since the end 2017.

What do developers value in open source?

- Developers appreciate collaborating and interacting with the open-source community more than contributing to open-source projects.
- Western European developers value almost every aspect of open source more than developers in other regions.
- South Asian developers highly value contributing to open-source projects, positioning this region to drive the next wave of open-source development.

Why do developers adopt or reject cloud technologies?

- Pricing and support/documentation dominate developers' decision-making process when adopting a cloud technology, but pricing is by far the most important reason for rejection.
- There are lots of opportunities for vendors to differentiate their orchestration tools in the market - developers are less concerned with price and more with features that help them to develop.
- Developers aren't so concerned with feature set or performance - as long as cloud solutions meet minimum requirements.
- Developers will reject technologies that don't provide them with a fulfilling development experience. Having access to a community and adequate support are important.

Emerging technologies

- Emerging technologies such as AR and VR are not fully engaging OSS principles.
- Little change in engagement and adoption rates indicate that DevOps has reached maturity.
- Fog/edge computing is gaining traction amongst developers engaged with the topic, but overall engagement is low.
- Computer vision is on the path to maturity, and with a strong pipeline of developers learning about this topic, adoption will continue to rise.
- Fatigue is setting in for some advanced technologies, but adoption is rising amongst developers who continue to engage.



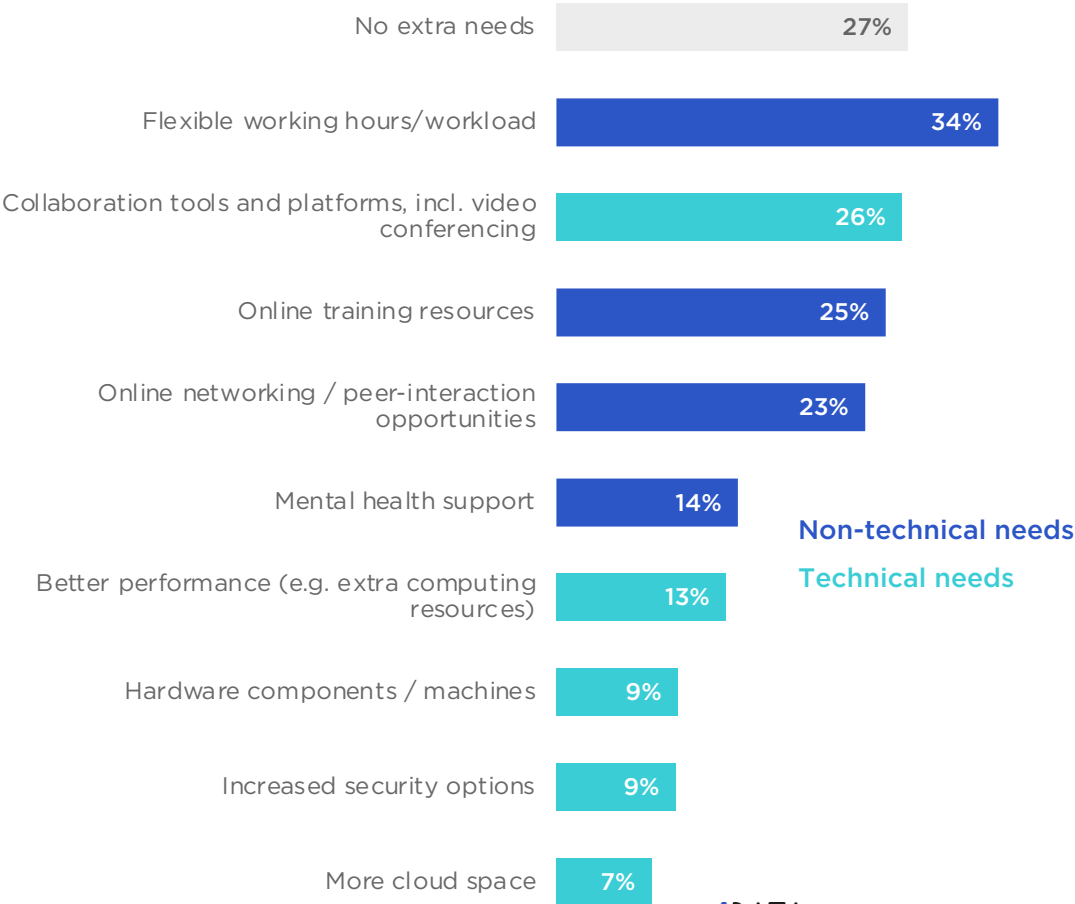
DEVELOPERS' EXTRA NEEDS DUE TO COVID-19

Working and performing during a pandemic will leave deep marks behind, both financially and psychologically speaking. In this chapter, we explore COVID-19's effects on developers' changing needs in relation to their development activities.

At the time of writing this chapter, there have been more than 30 million COVID-19 cases around the world, with 7.3 million of those still active. The virus is ubiquitous and affects all continents to more or less similar degrees. Working and performing during a pandemic is an experience that will undoubtedly leave deep marks behind, both financially and psychologically speaking. In this edition of our State of the Developer Nation report, we explore COVID-19's effects on developers' changing needs in relation to their development activities.

Flexibility in working hours and workload is the most pressing COVID-related need

% of developers (Q3 2020 n=16,113)



/DATA

7.2 MILLION DEVELOPERS REPORT NEEDING FLEXIBLE WORKING HOURS/WORKLOAD

We asked respondents to select from a given set of technical and non-technical needs, up to three extra needs the pandemic has created for their own development activities. 73% of developers reported having additional needs due to COVID-19. In particular, 34%, or 7.2 million developers, expressed their need for flexible working hours/workload.

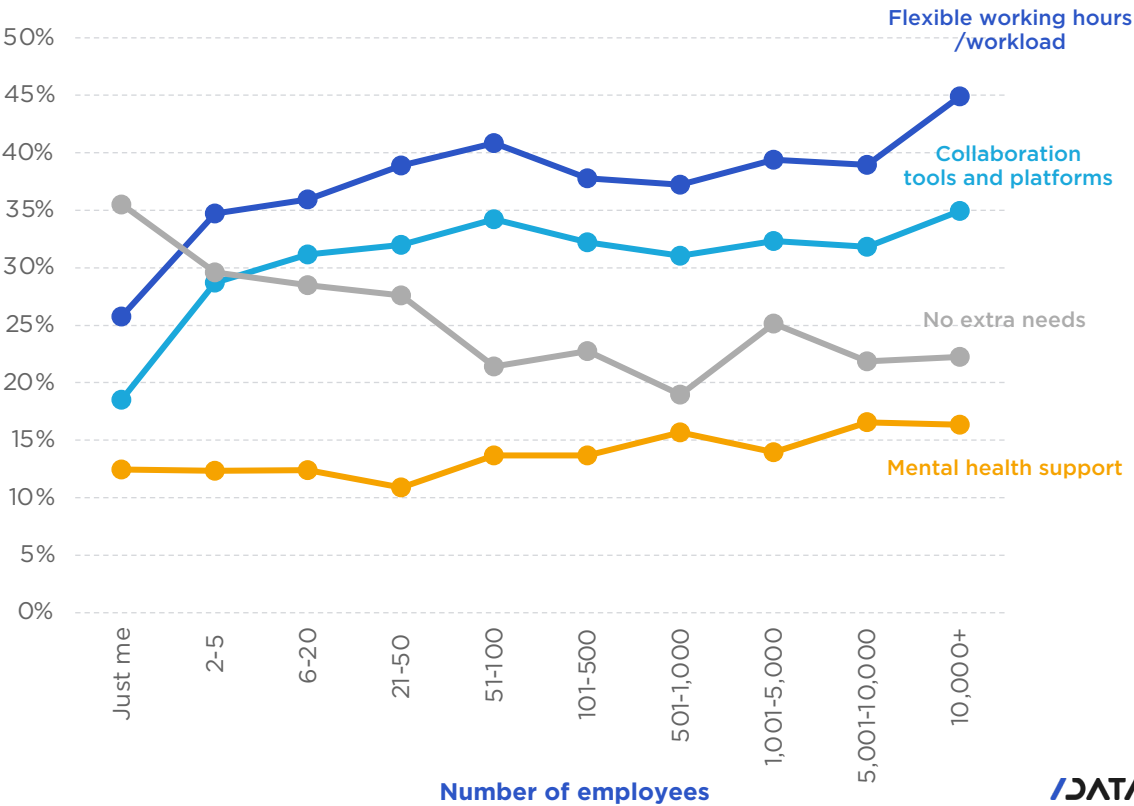
Quarantine and social distancing policies have encouraged many employers to allow their workers to work from home, where possible. A large proportion of workers are now facing the inconvenience of relocating their working space into their home. Among such inconveniences is the necessity of taking care of households while keeping up productivity. Under these circumstances, flexibility is seen as the key to success, or simply survival.

The next most common perceived needs, reported by about one in four developers, are: collaboration tools and platforms (26%), online training resources (25%), and virtual opportunities to support networking and peer-to-peer interaction (23%). Among these three, the only technical one, strictly speaking, refers to the need for collaboration tools, such as video conferencing platforms. The other top needs are related to self-improvement and self-management, and to socialising.

The supremacy of non-technical needs is striking. All of the technical necessities, except collaboration tools, sit at the bottom of the list, being reported only by about one in ten developers: better performance in terms of computing resources (13%), hardware components (9%), increased security (9%), and additional cloud space (7%). There are two explanations for these patterns. First, developers may have not indicated the need for extra technical support because it had been already fulfilled, i.e. their employers had already provided them with it. It could also be, however, that developers did not perceive technical considerations as being more important than flexibility, networking, and learning.

The bigger the company, the more flexibility is needed

% of developers, (Q3 2020 n=11,690)



We ran a random forest model and found that the most important factor in influencing developers' needs in relation to COVID-19 is their company size. Compared to those in middle- or large-sized companies, self-employed developers and developers working in small businesses of up to 20 employees report fewer new needs overall. That is especially the case for flexibility in terms of working hours/workload, and for collaboration tools. The most probable explanation is that they would have already implemented a flexible working schedule prior to COVID-19. This is likely to apply to contractors as well as to small, dynamic startups. When it comes to keeping collaboration and interaction going, it may just be easier for small groups of people to maintain old habits or find an easy-to-use tool, such as emailing, phoning, or even getting together whilst respecting the required social distancing.

On the contrary, the bigger the company, the stronger the need for all of the above, including opportunities for virtual interactions. A large company typically requires a structured system of communication, and usually that system needs to accommodate the various teams' diverse needs; even more so when a company is locked into an IT vendor's services.

Interestingly, the need for mental health support also linearly increases with company size, probably as a result of those challenges experienced in terms of flexibility and peer-to-peer communication and interaction. Another potential reason is that employees in larger organisations, where nobody is indispensable by default, may be experiencing more performance pressure and be more scared of losing their jobs.

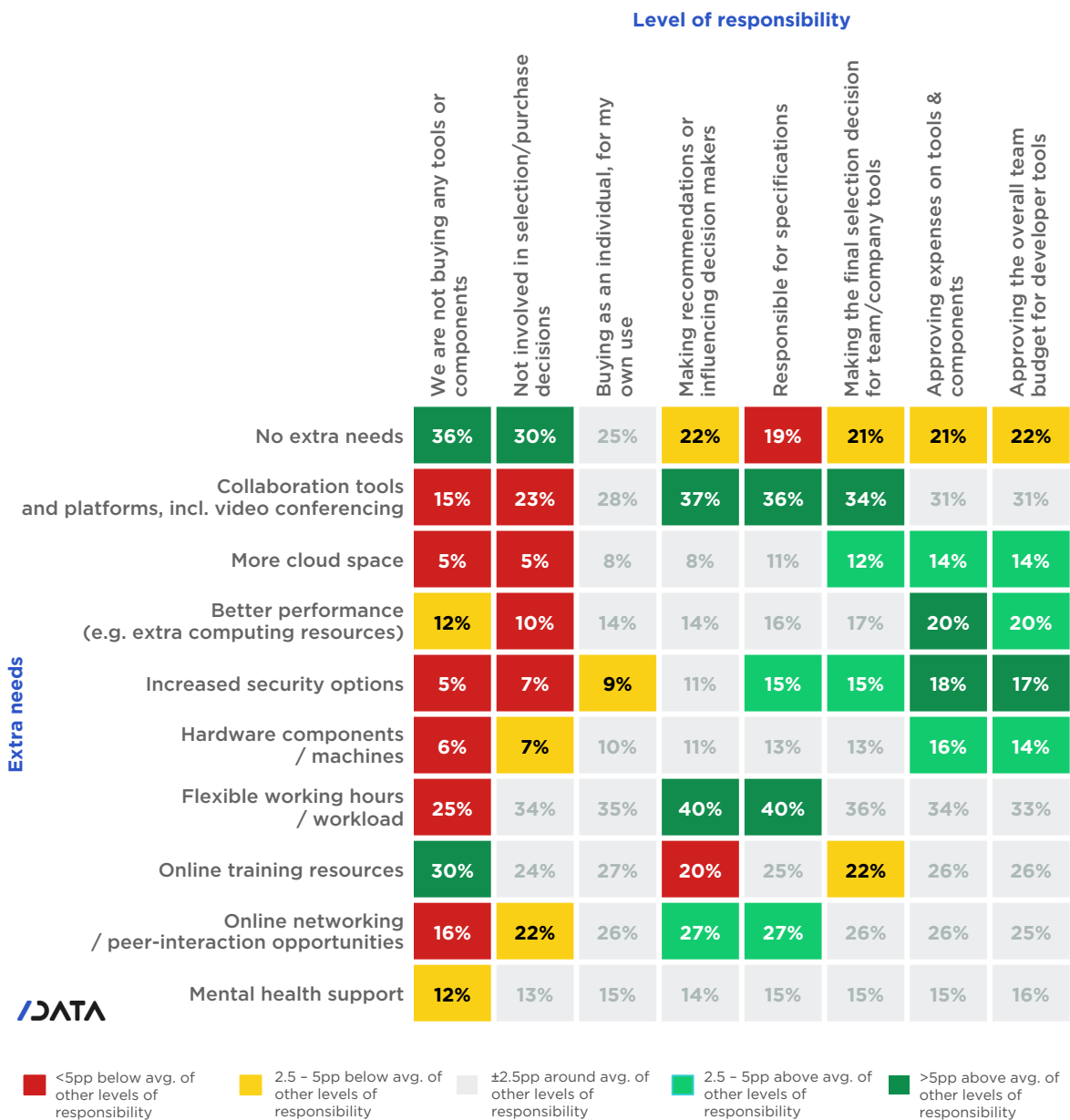
While developers' technical needs due to COVID-19 do not change significantly with company size, they strongly correlate to the developers' level of involvement in tool purchasing decisions. Those most concerned about increased security, performance, and cloud space are the ones responsible for tool specs and expenses, as well as budget approval, who usually fulfill roles within technical management.

On the one hand, with the increasing number of developers working from home, more machines need to be available and connected via VPN and similar technologies. More layers to navigate introduces complexity barriers that affect work efficiency, but also the need for the implementation of extra security controls. Furthermore, servers are often overloaded and downtimes happen more frequently, affecting system reliability. If you add to this the fact that budgets are being reduced or even frozen, due to the economic instability the pandemic is causing, the situation is actually precarious. Those in charge are inevitably the ones noticing the need for technical support the most.

In a relatively short time, the pandemic has generated and consolidated a series of working practices that had been previously known only to a very small proportion of the population. Such new practices, based on remote working and virtual collaboration, are likely to persist after COVID-19. If one acknowledges this, investing in optimising support becomes even more valuable. We recommend that, especially large enterprises, consider the delicate balance between self-management and collaboration needs when designing policies and offering support to their employees in the face of the pandemic situation.

Technical needs are mostly felt by influencing decision-makers

% of developers per level of responsibility in tool buying decisions
(Q3 2020 n=15,987)





PROGRAMMING LANGUAGE COMMUNITIES – AN UPDATE

The choice of programming language matters deeply to developers because they want to keep their skills up to date and marketable. Languages are a beloved subject of debate and the kernels of some of the strongest developer communities. They matter to toolmakers too, because they want to make sure they provide the most useful SDKs.

It can be hard to assess how widely used a programming language is. The indices available from players like Tiobe, Redmonk, Stack Overflow's yearly survey, or GitHub's Octoverse are great, but offer mostly relative comparisons between languages, providing no sense of the absolute size of each community. They may also be biased geographically, or skewed towards certain fields of software development or open-source developers.

Size of programming language communities in Q3 2020

Active software developers, globally, in millions (Q3 2020 n=11,927)

		Most popular in	Least popular in
Javascript*	12.4M	Web, Cloud	DS/ML, AR/VR
Python	9.0M	DS/ML, IoT apps	Mobile, Web
Java	8.2M	Mobile, Cloud	DS/ML, Web
C/C++	6.3M	IoT apps, IoT devices, AR/VR	Web, Cloud, Mobile
PHP	6.1M	Web, Cloud	DS/ML, Mobile
C#	6.0M	Games, AR/VR, Desktop	DS/ML, Mobile
Visual development tools	2.8M	Desktop, AR/VR	Cloud, Web
Swift	2.4M	Mobile, AR/VR	Cloud, IoT devices
Kotlin	2.3M	Mobile, AR/VR	DS/ML, Desktop
Go	1.5M	Cloud, AR/VR	DS/ML, Web
Ruby	1.5M	IoT apps, Cloud	DS/ML, apps for 3rd party ecosystems
Objective C	1.4M	AR/VR, Mobile	Desktop, IoT devices
Rust	0.8M	AR/VR, IoT apps	Web, Cloud
Lua	0.8M	AR/VR, Games	Mobile, apps for 3rd party ecosystems

(*) JavaScript includes CoffeeScript and TypeScript

The estimates we present here look at active software developers using each programming language; across the globe and across all kinds of programmers. They are based on two pieces of data. First, our independent estimate of the global number of software developers, which we published for the first time in 2017. We estimate that in mid-2020 there were 21.3M active software developers in the world. Second, our large-scale, low-bias surveys which reach tens of thousands of developers every six months. In the surveys, we consistently ask developers about their use of programming languages across ten areas of development, giving us rich and reliable information about who uses each language and in which context.

Nearly 5M developers joined the JavaScript community in the last three years

JavaScript is the most popular programming language by a wide margin, with 12.4M developers globally using it. Notably, the JavaScript community has been growing in size consistently for the past three years. Between Q2 2017 and Q3 2020, nearly 5M developers joined the community - by far the highest growth in absolute terms across all languages. Even in software sectors where JavaScript is least popular, like data science or AR/VR, over a fifth of developers use it in their projects.

For the second half-year period in a row, Python is the most widely adopted language behind JavaScript. Python now counts 9M users, after adding 2.2M net new developers in the past year alone, outranking Java at the beginning of 2020. The rise of data science and machine learning (ML) is a clear factor in its popularity. An impressive 77% of ML developers and data scientists currently use Python. For perspective, only 22% use R, the other language often associated with data science.

Java, with over 8M active users worldwide, is the cornerstone of the mobile app ecosystem - Android - as well as one of the most important general-purpose languages. It's adoption may have remained stable in the past six months but, in the overall picture, the Java community has gained 1.6M developers since mid-2017, which corresponds to a 24% growth.

The group of major, well-established languages is completed with C/C++ (6.3M), PHP (6.1M) and C# (6M). The fact that C# lost three places in the ranking of language communities during the last three years is mostly explained by its slower growth compared to C/C++ and PHP. C and C++ remain core languages in IoT projects (for both on-device and application-level coding), whereas PHP is still the second most commonly used language in web applications, after JavaScript. On the other hand, C# may be sustaining its dominance in the game and AR/VR developer ecosystems, but it seems to be losing its edge in desktop development - possibly due to the emergence of cross-platform tools based on web technologies.

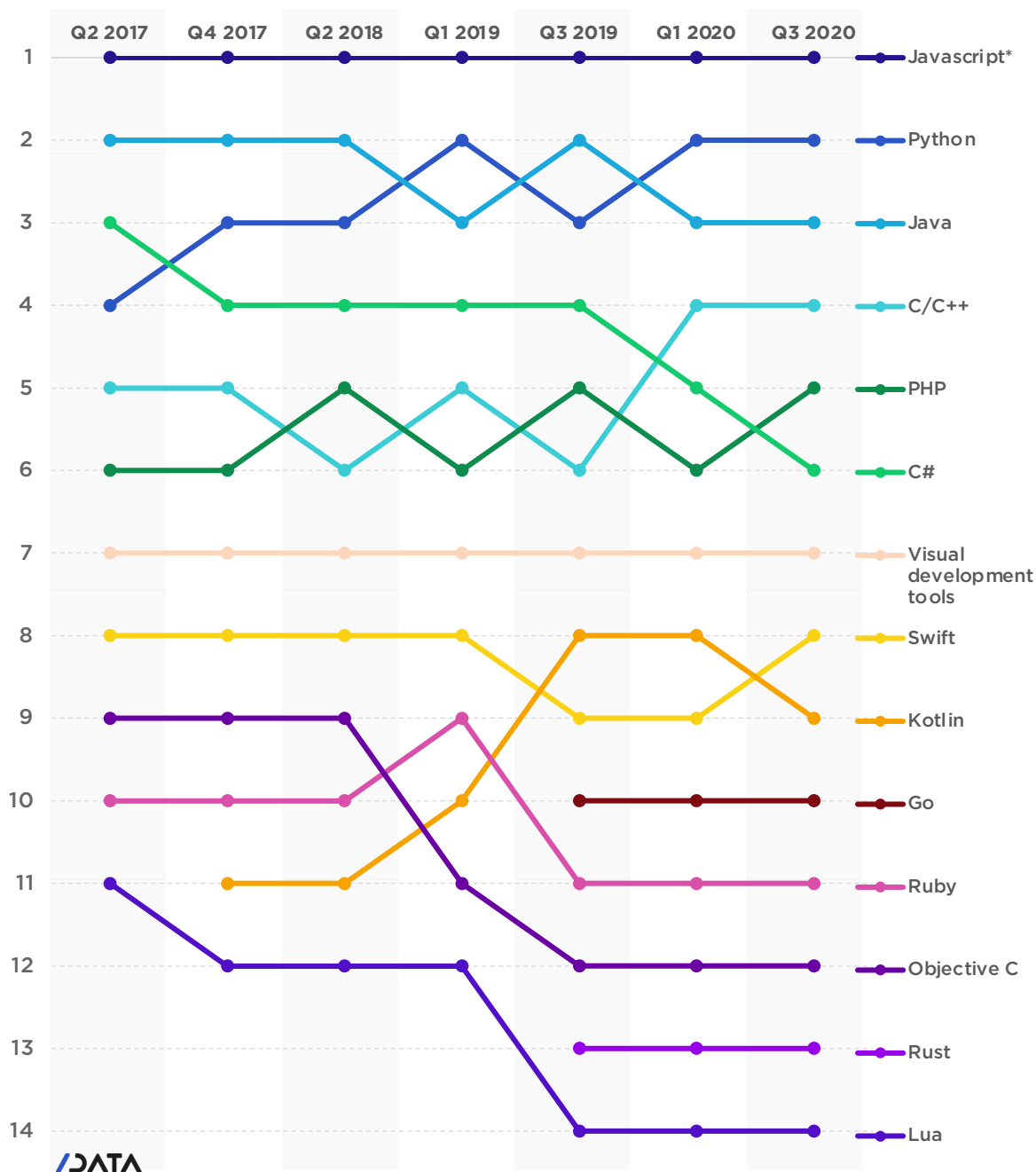
“ C# is growing in popularity, but at a slower pace than PHP and C/C++ ”

As we had found in previous editions of this report too, Kotlin is one of the fastest growing language communities, having increased more than two-fold in size since the end of 2017, from 1.1M in Q4 2017 to 2.3M in Q3 2020. This is also very evident from Kotlin's ranking, where it moved from 11th to ninth place during that period - a trend that's largely attributed to Google's decision to make Kotlin its preferred language for Android development.

Swift surpassed Kotlin in popularity this year, after attracting slightly more net new developers in the first half of 2020 (400k vs 300k). Since Swift became the default language for development across all Apple platforms, the adoption of Objective C has been decreasing steadily. This phase-out from the Apple app ecosystem is also matched by a significant drop in the rank of Objective C, from ninth to 12th place.

Finally, the more niche languages - Go, Ruby, Rust, and Lua - are still much smaller, with up to 1.5M active software developers each. Ruby and Lua have been around for more than two decades now, but their communities have essentially stopped growing in the last three years. On the contrary, Go and Rust appear to be actively adding developers, although it is still unclear whether the two languages will climb the programming language ranking in the coming period.

Rank of programming language communities, 2017-2020



(*) JavaScript includes CoffeeScript and TypeScript



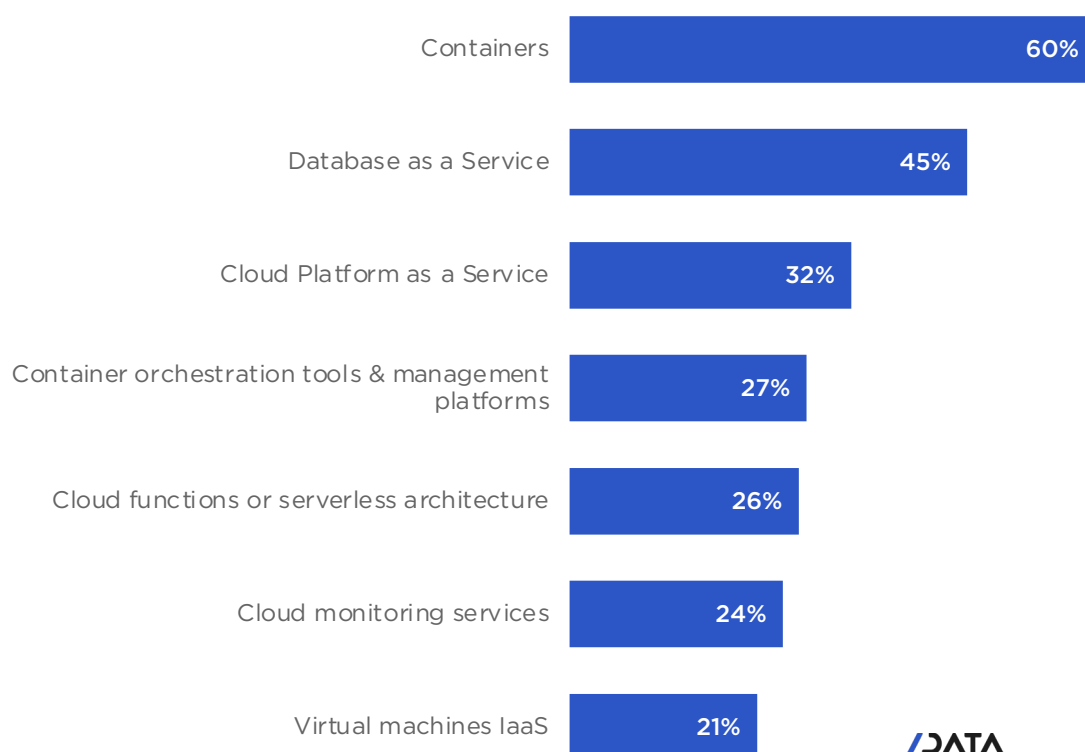
WHY DO DEVELOPERS ADOPT OR REJECT CLOUD TECHNOLOGIES?

In the nearly fifteen years since Amazon AWS cracked open the cloud market by releasing S3 - and changed the world by doing so - there has been huge growth in the variety of cloud solutions available for developers to use. In this chapter, we'll examine the different reasons that developers give for adopting or rejecting cloud technologies.

Of the new cloud technologies which have appeared over the last fifteen years, containers have arguably had the greatest impact. With 60% of developers using this technology, the benefits are clearly widely recognised. However, with just under 30% of developers using container orchestration tools and management platforms, there is still room for this technology to develop. In second position, with 45% of cloud developers using this technology, Database-as-a-Service (DBaaS) is also very widely used, and data storage and retrieval will continue to be an important issue, albeit in a much more sophisticated form than S3 originally offered. Cloud Platform-as-a-Service (PaaS) sits in a distant third place. A third of backend developers are using PaaS, putting this technology slightly ahead of the other ones we ask about - between 21-27% of developers use them.

Containers is the cloud technology that is most widely used by backend developers

% of backend developers using cloud technologies (Q3 2020 n=3,978)



Abstraction and simplification are two of the main drivers for the mass adoption of cloud technologies, but we can't overlook the role that flexibility plays. Spinning up instances to cope with variable demand, creating temporary testing environments, and adding storage as required is immensely powerful. But one often-overlooked aspect of this flexibility is that developers and organisations have the flexibility to choose. They are not restricted to the expensive, bare metal they bought ten years ago, and they are less constrained by monolithic purchasing processes, because, to put it simply, these decisions matter less. In a world where infrastructure can be provisioned and destroyed at will, and where data and server configurations can be transferred easily between homogeneous systems, cloud providers have to find other areas of differentiation in order to compete. Vendor lock-in is much less of an issue for users than it once was, and the rise of the developer as a decision-maker has put even more power into their hands. Note the adoption and rejection reasons for DBaaS and orchestration platforms come from our previous survey, fielded in Q1 2020.

Pricing and support/documentation are most important to developers

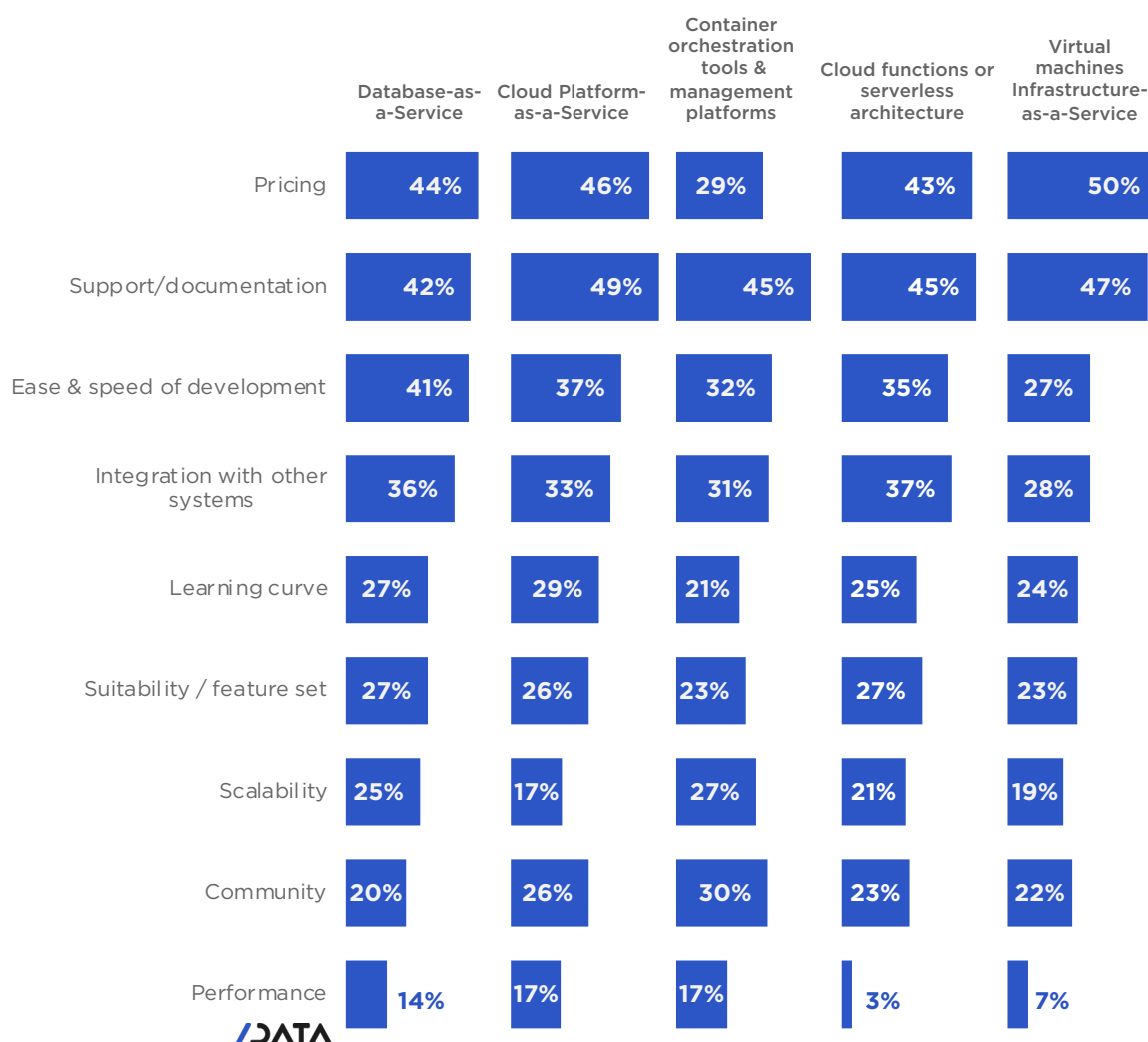
For every cloud technology, with the exception of orchestration tools, pricing and support/documentation are the two most important factors that developers consider when adopting that technology. For the most part, these two factors switch between first and second place, however, pricing drops to fifth place for developers considering adopting an orchestration tool, whereas support/documentation remains at the top by a large margin. Around three in ten of these developers selected ease and speed of development (32%), integration with other systems (31%), community (30%), and pricing (29%) as reasons for adoption, with pricing being around 15 percentage points lower for orchestration tools than for other technologies. On the other hand, community and scalability are generally more important for developers selecting an orchestration tool.

Much of this distinction is driven by the dominance of Kubernetes. With 57% of backend developers who are using an orchestration tool choosing Kubernetes, it is the single most popular orchestration tool, and importantly, it's free and open source. It stands to reason, therefore, that pricing is simply not an issue for developers using Kubernetes, instead they value the community support that helps them master such a complex tool.

Indeed, as well as pricing being much less important for these developers, the learning curve is also less important. It seems that these developers understand that they are dealing with high levels of complexity and abstraction and accept that there is a lot to learn in this space. But for those developers that want the abstraction and simplicity offered by a commercial container management system, many paid options exist, and pricing is still an important factor in this space.

Ranking of reasons for adoption

(Q3 2020 n=1,603 | Q1 2020 n=1,935)



Taking developers' reasons for rejection into account lets us view the decision-making process from the other side. Immediately, we see that pricing is the dominant factor when rejecting every technology. Taking a closer look at the data shows the true extent of this - for DBaaS and Infrastructure-as-a-Service (IaaS), developers were more than twice as likely to select pricing as a rejection reason than the second- and third-placed reasons of support/documentation, and the learning curve, respectively. Amongst the remaining technologies, the smallest difference was 8 percentage points, for developers rejecting orchestration tools. Further down the list, there is a lot of variability between the different technologies. For example, the learning curve was the second most popular rejection reason for developers choosing IaaS, with a quarter of them doing so. This suggests that the learning curve for IaaS is quite steep and that this is a barrier for many developers. This is not the case for DBaaS however, where only 15% of developers stated this as a reason for rejection.



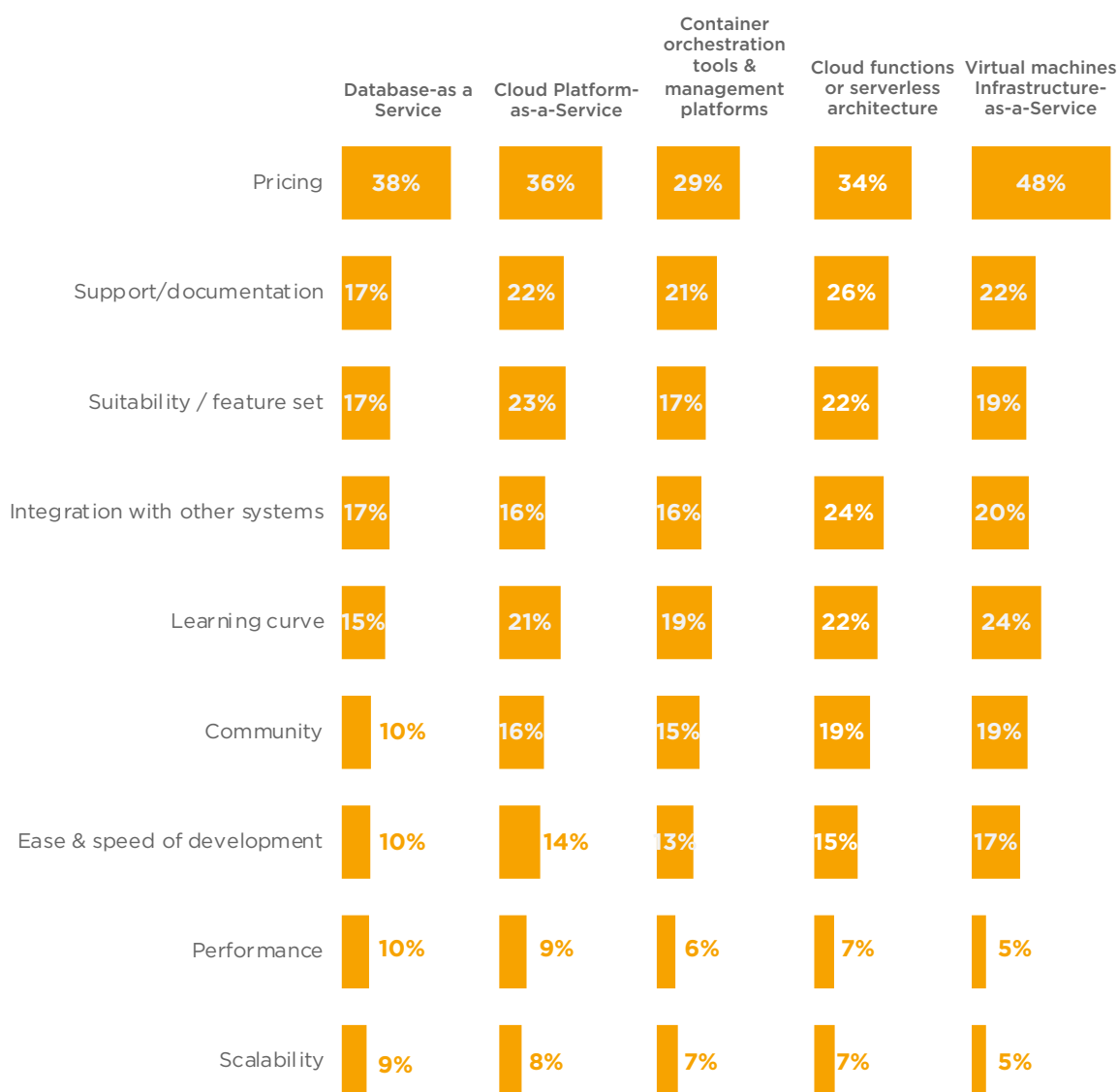
Suitability, feature set, and performance are hygiene factors

Suitability and feature set has middling importance for developers choosing to adopt a technology, but for many technologies, it is a more important reason for rejection. This shows that suitability and feature set is a hygiene factor - there are relatively few cases where this is of paramount importance, but many where a technology does not meet the needs and is therefore rejected.

Finally, performance sits very low in the hierarchy for developers adopting and rejecting cloud solutions. This indicates that, for the vast majority of uses cases, the range of performance options provided by vendors is sufficient. This suggests that many cloud computing products are, to some extent, homogenous, and that developers are more concerned with the 'soft' features, such as support/documentation, community, or learning curve. These features make for a fulfilling development experience, and in the age of the developer as a decision-maker, experience is everything.

Ranking of reasons for rejection

(Q3 2020 n=932 | Q1 2020 n=1,135)





WHO IS INTO DEVOPS?

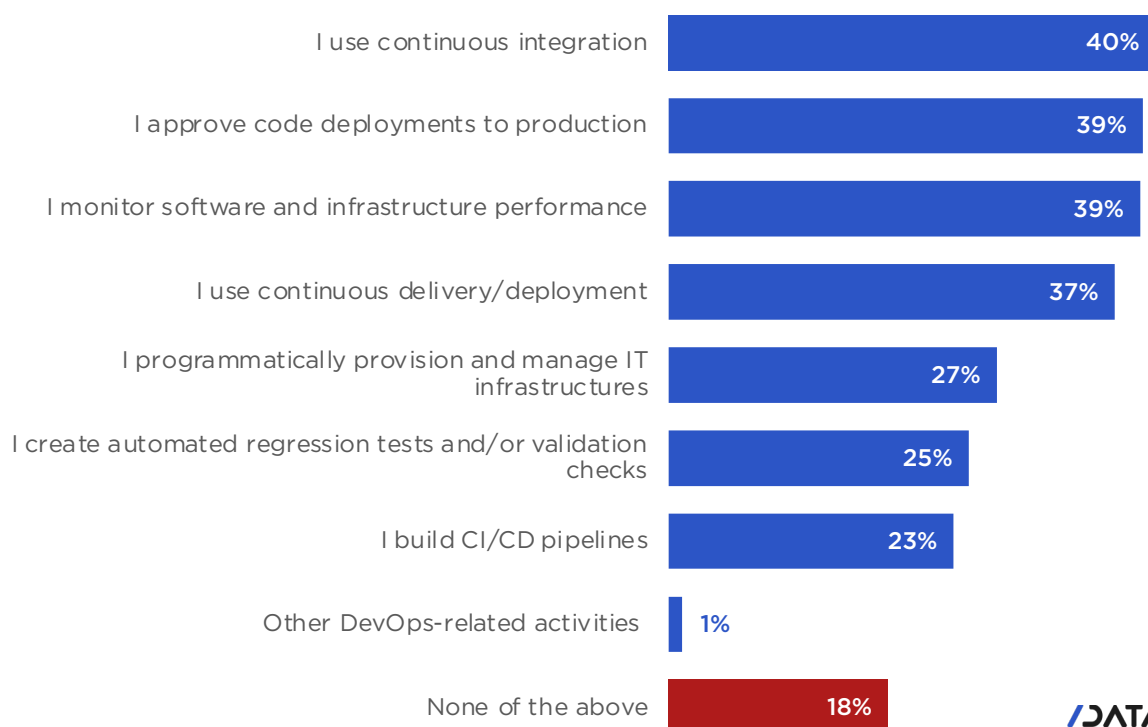
The term “DevOps” is typically used as a catch-all phrase to describe a cultural shift within organisations that closes the gap between software development and IT operations teams, which historically functioned independently. In essence, DevOps is a set of tools and practices that seek to automate and streamline the software development and release process. But who exactly are the DevOps practitioners within the global developer ecosystem and what activities are they involved in? Are there specific roles or software sectors that are particularly associated with the DevOps culture?

To answer the above questions, we asked developers in our latest Developer Economics survey whether they are involved in any of the activities that commonly fall under the DevOps spectrum, ranging from continuous integration and deployment to application and infrastructure monitoring. For the purposes of this chapter, we only consider developers who are professionals in at least one of the software areas they are active in.

The first thing to note is that the adoption of DevOps practices is widespread among professionals, perhaps even more so than one might expect, given that the DevOps movement is relatively new. According to our data, the vast majority of professional developers (82%) are involved in DevOps in one way or another. For perspective, just over half (52%) of non-professionals are involved in any of the DevOps activities on our list.

Which of the following development activities are you involved in?

% of professional developers (Q3 2020 n=11,688)



“ *The vast majority of professional developers are involved in DevOps, but do not necessarily consider themselves DevOps practitioners* ”

On a separate view of engagement with DevOps in our survey, only one in five developers reported working on DevOps when they were explicitly asked about their involvement in several emerging areas, including blockchain applications and quantum computing, among others. Even if we include those who said that they are learning about or are interested in DevOps, no more than 65% consider themselves to be engaged with the area. This signals that there is a large portion of the developer population that have already adopted DevOps practices but do not necessarily self-identify with the term.

Focussing on the individual steps of the DevOps lifecycle, we find that developers are first and foremost involved in the fundamental activity of releasing frequent but small software updates. The most popular development process related to DevOps is continuous integration (CI), practiced by 40% of respondents. Another 37% use continuous delivery or deployment (CD), which expands upon CI by automatically deploying all code changes to staging or production environments.

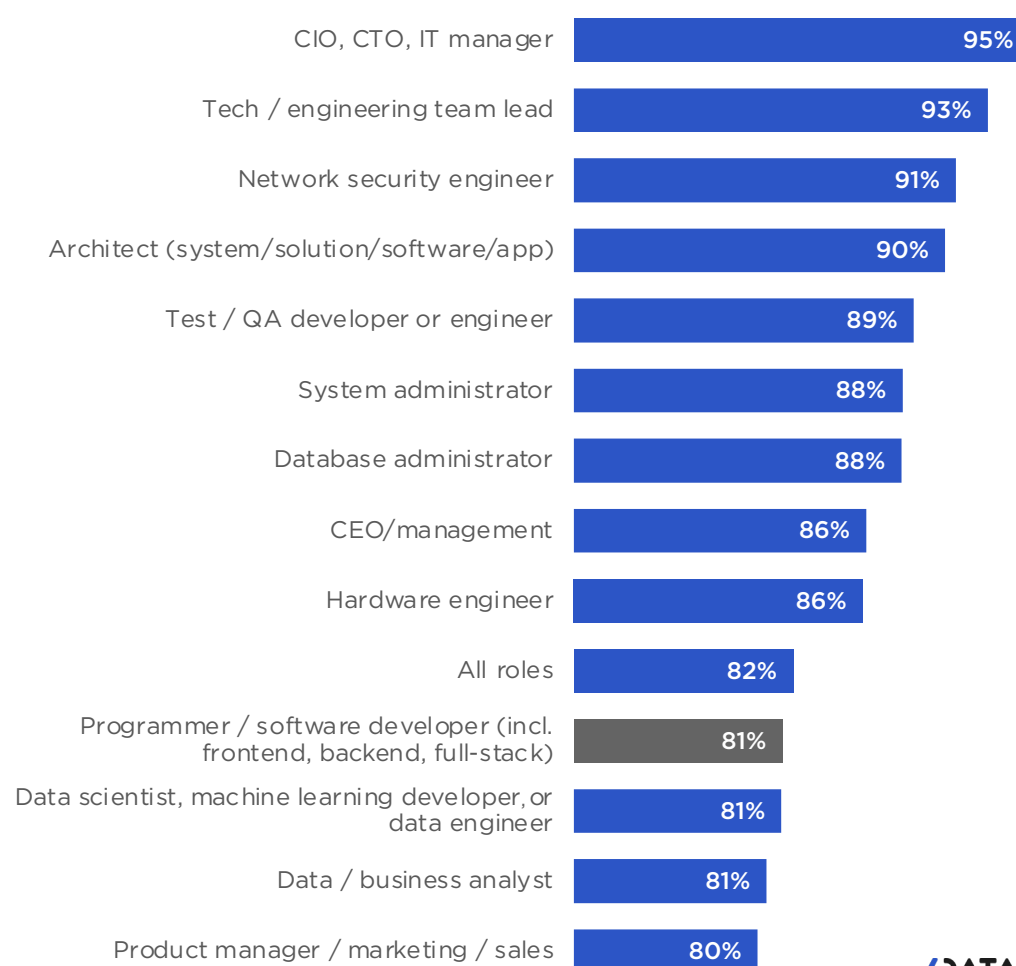
However, full automation of the software release process - and therefore true commitment to the DevOps culture - is far from a reality. While more than half (52%) of developers use CI or CD to streamline parts of their workflow, only 25% use both practices to automate all steps between integrating code changes into a central repository through to production deployment. As it turns out, developers are still sceptical about fully automated CI/CD pipelines. This is evident by the fact that nearly 40% of them manually give the green light for code deployments to be promoted to production.

Application and infrastructure monitoring, performed by 39% of developers, is one of the most common development practices, but not so much infrastructure provisioning and management (27%), which is still the realm of IT managers and system administrators. Similarly, creating automated tests (25%) and building CI/CD pipelines (23%) are rather specialised tasks, carried out predominantly by quality assurance professionals and solution architects, respectively.

Talking about organisational roles; our research reveals noticeable differences in the level of DevOps adoption, i.e. involvement in any DevOps-related activity, depending on the title that developers hold. First of all, technical company leaders - CIOs, CTOs, IT managers, and engineering team leads - report the highest level of involvement in DevOps activities. Not only do almost all developers with a technical leadership function, about 95% of them, have at least some participation in the DevOps lifecycle, but they are also simultaneously involved in a higher than average number of DevOps activities (three vs two).

Involvement in DevOps, by company role

% of professional developers (Q3 2020 n=11,687)





Programmers have largely adopted CI/CD processes, but not so much other DevOps practices

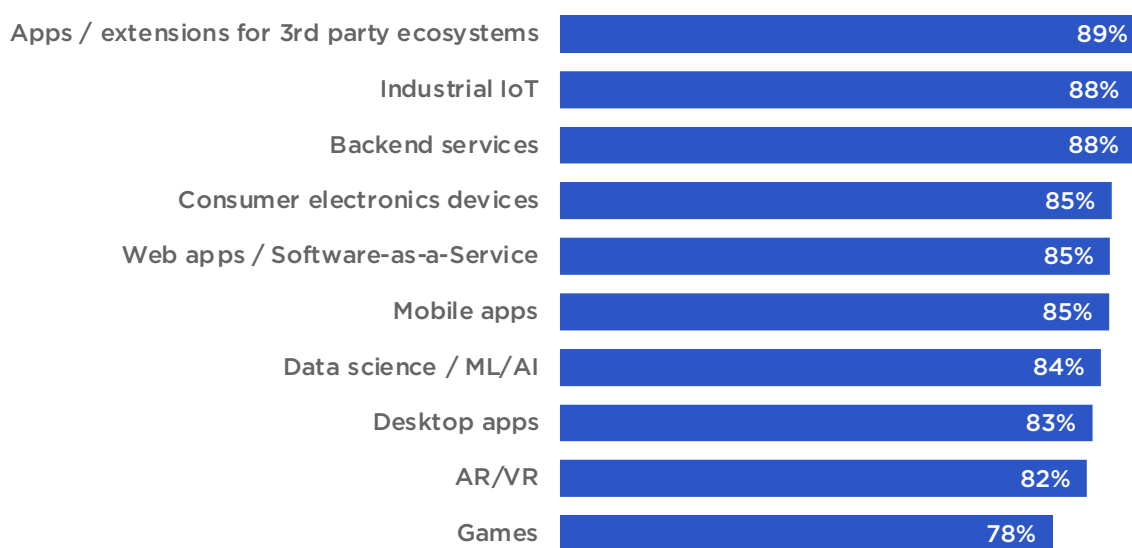
The next tier of the DevOps adoption ranking is mainly occupied by specialist roles, such as network security engineers, QA developers, and system administrators. Between 86% and 91% of developers holding these positions are in some way associated with the DevOps culture. We should note, however, that only architects - system, solution, software etc. - appear to be heavily involved in all phases of the DevOps lifecycle. All other specialists are primarily focussed on activities relevant to their expertise. For example, system administrators are naturally focussed on infrastructure provisioning and monitoring, whereas QA engineers create automated tests for CI/CD pipelines more than anything else.

Front-line coders and software developers, who represent the majority of respondents in our survey (61%), are also highly likely to be involved in DevOps activities - 81% of them are although not more often than the average professional (82%). Our data suggests that software developers are keen to adopt CI/CD processes, but not so much operational practices such as monitoring applications in production environments. Again, this indicates that the complete shift to the DevOps culture has not yet been achieved. Apart from responsibilities central to their role, programmers are not accountable for additional product lifecycle phases.

Another important indicator of the level of engagement with DevOps practices is the software sectors that developers are involved in. As with roles, we see some interesting variations in DevOps adoption across sectors. For example, close to 90% of developers who create extensions for third party ecosystems or backend services are into DevOps, as opposed to less than 80% of game developers.

Involvement in DevOps, by software sector

% of professional developers (Q3 2020 n=11,688)



That is partly explained by the extensive coding experience required to implement the DevOps model. We know from our data that DevOps practitioners are far more experienced coders than developers who are not involved in any DevOps-related activity. And developers working on apps for third party ecosystems, backend services, or industrial IoT projects are among the most experienced in the software economy: up to 85% of them have three or more years of coding experience. In comparison, no more than 73% of game developers have the same level of expertise.

Nonetheless, we find that desktop app developers report relatively low adoption of DevOps practices, even though they are highly experienced professionals - 82% of them have at least three years of experience in software development. This points to limited alignment with the key benefits of DevOps more than anything else. Desktop applications typically receive updates at a lower frequency than applications running on other environments, e.g. servers. Therefore, the fundamental DevOps strategy of releasing small software updates at high velocity is not entirely applicable to desktop application projects.

In conclusion, DevOps signifies a cultural shift whereby developers from different teams work closely together with an aim to deliver software faster and more reliably. The practices of the DevOps model are already widely adopted among professional developers across software sectors and organisational roles, although with some significant variations in the focus on specific activities. These variations reveal, in some cases, that true commitment to the DevOps culture is not yet achieved; many developers are still focussed on the core aspects of their role instead of assuming responsibility for additional phases of the product life cycle.

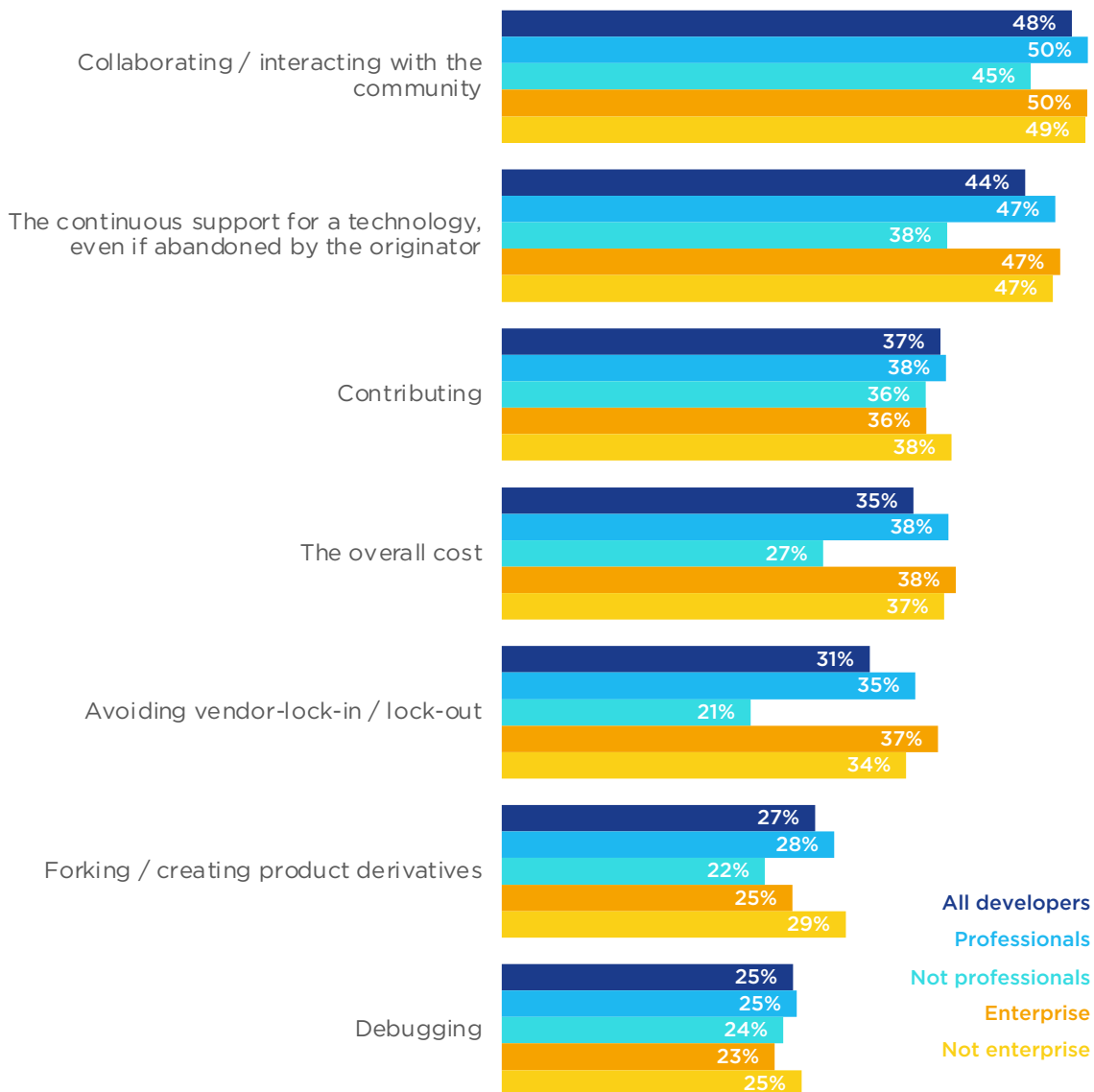


WHAT DO DEVELOPERS VALUE IN OPEN SOURCE?

Open-source software (OSS) is used by 92% of developers, so what exactly do they value in it? We find that developers value OSS's ability to supersede any single contributor and live on almost eternally. We highlight some uncertainty around OSS's future by showing trends from geographic regions and sectors.

What developers value in using open source software

% of developers who agree with the statement (Q3 2020 n=15,064)



/DATA

Note: Professional developers (72% of the sample) self-identify as working in at least one of the eleven sectors that we research. Non-professional developers (28% of the sample) instead self-identify as either hobbyists or learners. Enterprise developers (17% of the sample) are developers who self-identify as working in organisations which exceed 1,000 employees.

Open-source software (OSS) is ubiquitous in the global developer community. As our data shows, OSS is used by 92% of developers. A question that comes to mind is: what exactly do developers value in OSS? In the chart below, we show which statements developers value about OSS, broken down by professional and non-professional developers, and enterprise and non-enterprise developers.

The overarching theme for what developers value from OSS is its ability to be eternal. “To collaborate with the community, building software that outlasts even its originator” encapsulates the two statements with the greatest agreement.

***“ Nearly half of developers value OSS because it supports collaborating /interacting with the community.* ”**

The overall cost and wanting to avoid vendor lock-in/lock-out are important aspects that professional and enterprise developers in particular value in OSS, while non-enterprise developers value forking product derivatives and debugging more than the other groups. Non-professional developers do not value the overall costs element, perhaps because they have not experienced the costs involved in closed-source software, whereas many professional developers have. Another aspect that non-professional developers value significantly less is avoiding vendor lock-in. This also suggests that these developers have not experienced the limitations of closed-source software yet.

Appreciation of the overall costs of OSS is also highly linked with years of developer experience: only 24% of developers with less than one year of experience agree that low cost is an asset of OSS. In contrast, the percentage of developers who agree that low cost is an asset of OSS rises to 34% of developers who have between three and five years, and 43% of developers with six or more years of experience. Typically, as developers gain experience, they begin to work in different sectors, often crossing over between sectors. At this point, the flexibility that OSS offers may become crucial.

Finally, we also see a greater proportion of non-professional developers not using OSS compared to others. This is also reflected indirectly in each of the other statements; we see that non-professional developers agree with every statement less than professional developers. This suggests that, to be truly appreciative of the benefits of OSS, you may have had to engage with it seriously, in the way professional developers do.

Where OSS is written is changing

At present, the culture of OSS is particularly strong with Western European and Israeli developers, where not a single statement is valued below the average. On the contrary, developers in North America—who, up until now, have driven the OSS movement—value contributing and interacting with the community less than average. This could suggest a cooling off of North American OSS development and a maturing of this ecosystem.

On average, East Asian developers seem to be disengaged from the OSS movement more than developers from other regions. Only 88% of developers in this region use OSS compared to 92% globally. In general, developers in this region also value less aspects of OSS. In particular, their extremely low appreciation of the continuous support for the technology compared to others, highlights that developers in this region are apprehensive about the longevity of OSS, which partially undermines its main benefit. This apprehension is also reflected by the relatively low agreement associated with contributing.

According to our data, South Asian developers value contributing to OSS significantly more than others. In addition, South Asia is the region with the largest proportion of developers who value collaborating and interacting with the community. This combination positions the region to be among the drivers of the next wave of OSS development.

In the Middle East and Africa region, some key advantages of OSS, such as avoiding vendor lock-in and the overall low cost have not yet resonated with developers — this is despite the fact that, at least for Africa, [income per capita is low compared to global averages](#). What assists in explaining this is this region's proportion of professional developers and the experience of its developers. The Middle East and Africa, as well as South America, have roughly the same proportion of professional developers, 60.7%, in contrast to North America or Western Europe and Israel, where more than 80% of developers are professional. Non-professionals value OSS less. Similarly, developers in the Middle East and Africa are also the least experienced, on average, and years of experience in particular is linked with appreciating the low cost of OSS.

What developers from different regions value in OSS

% of developers in each region who value each aspect of OSS. Regions arranged by size of developers population (Q3 2020 n=15,064)

	Regions								
	All regions	North America	Western Europe & Israel	East Asia	Middle East & Africa	South Asia	Eastern Europe, Russia & Former CIS	South America	Oceania
Collaborating / interacting with the community	48%	44%	53%	38%	51%	55%	47%	54%	51%
The continuous support for a technology, even if abandoned by the originator	44%	46%	51%	29%	43%	42%	49%	48%	59%
Contributing	37%	33%	38%	29%	42%	50%	34%	42%	34%
The overall cost	35%	41%	38%	34%	27%	28%	31%	32%	54%
Avoiding vendor-lock-in / lock-out	31%	33%	39%	24%	19%	16%	49%	33%	53%
Forking / creating product derivatives	27%	26%	30%	20%	23%	31%	35%	26%	30%
Debugging	25%	27%	24%	26%	27%	29%	19%	16%	24%

■ <5pp below avg. of all regions
 ■ 2.5 - 5pp below avg. of all regions
 ■ ±2.5pp around avg. of all regions
 ■ 2.5 - 5pp above avg. of all regions
 ■ >5pp above avg. of all regions

Some sectors embrace OSS while others don't

Emergent sectors such as augmented reality (AR) and virtual reality (VR) stand to benefit greatly from OSS as a means of defining a common standard and exchanging ideas. Yet, we find that developers working in these two fields do not value forking/creating product derivatives, nor even collaboration in the case of VR, as much as other developers do, on average, from other fields. This could be partially explained by the lower than average agreement with the need for continuous support for a technology. When developers do not value this characteristic, it is unlikely that they are working with the mindset which would ensure long term OSS growth and desirability.

On the other hand, developers who are building apps and extensions for third party ecosystems, on average, value contributing and forking more than developers in other sectors. Similarly, the very successful node.js runtime has facilitated other extensions and developers working in backend services really value the continuous support of OSS projects.

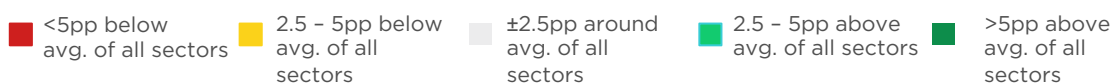
At present, despite the large percentage of developers who use open source software, it is only in certain circumstances that the majority of developers value OSS for any given reason. Perhaps this suggests that OSS has become an expectation rather than being perceived as a gift from society at large to society at large. Observing how developers value OSS in the future would be a good litmus test for the health of open source projects. For now though, there are encouraging blooms in South Asia for example, but also software sectors of scepticism, such as in AR/VR.

What developers involved in different sectors value in OSS

% of developers, who do use OSS, in each sector who value each aspect of OSS. Only rows of interest shown (Q3 2020 n= 15,064)

	Sectors					
	Average	Backend services	Augmented reality including non-developers	Virtual reality including non-developers	Consumer electronics devices	Apps / extensions for 3rd party ecosystems
Collaborating / interacting with the community	55%	58%	56%	51%	49%	59%
The continuous support for a technology, even if abandoned by the originator	49%	55%	45%	44%	46%	55%
Contributing	44%	44%	44%	45%	40%	48%
Avoiding vendor-lock-in / lock-out	35%	42%	31%	31%	35%	40%
Forking / creating product derivatives	33%	34%	30%	30%	34%	40%
Debugging	32%	28%	35%	38%	35%	31%

/DATA





EMERGING TECHNOLOGIES

As interest in a technology waxes and wanes, so does its influence. The hot topic of yesterday becomes insignificant in the face of new challenges and opportunities. Many factors influence the trajectory of this curve, and understanding developers' willingness to work with and learn about emerging technologies gives us a valuable insight into when and how these emerging technologies may change the world.

We have tracked developers' engagement with and adoption of different technologies over five surveys, spanning two-and-a-half years, ending Q3 2020. To measure engagement and adoption, we asked developers if they are working on, learning about, interested in, or not interested in different emerging technologies, whilst adding to the list as new innovations appear.

For the purpose of our analysis, we have defined developers engaged with a specific technology as those that have indicated they are either interested in, learning about, or working on it. We measured adoption as the share of developers who are engaged with a technology and have indicated they are working on it. To contextualise this information, we classified each technology according to whether its engagement rate is above or below the median - high/low engagement - and whether its adoption rate is above or below the median - high/low adoption. This allows us to put each technology into one of four quadrants:

- 1. High engagement/high adoption** - these technologies capture the imagination of many developers and have proven commercial success.
- 2. High engagement/low adoption** - these technologies capture the imagination of many developers, but have yet to make a commercial impact.
- 3. Low engagement/low adoption** - These fringe technologies do not interest many developers, and their commercial value is yet to be proven.
- 4. Low engagement/high adoption** - These technologies might not appeal to many developers, but for those that are interested, commercial adoption is high.

DevOps continues to see high engagement, with 60% of developers either working on, learning about, or interested in this technology. The commercial benefits from integrating DevOps and associated working practises, such as continuous integration/continuous deployment (CI/CD) are clear - more automation and a short product life cycle allows companies that adopt DevOps to do more, with less, especially for cloud-native organisations.

This demand has translated into elevated salaries for DevOps practitioners - at the time of writing, the [average salary in the US is close to \\$100k](#). This is a strong pull for developers looking to augment their skills, and almost three in ten of these engaged developers are actively learning about the topic, with the same amount having already adopted it.

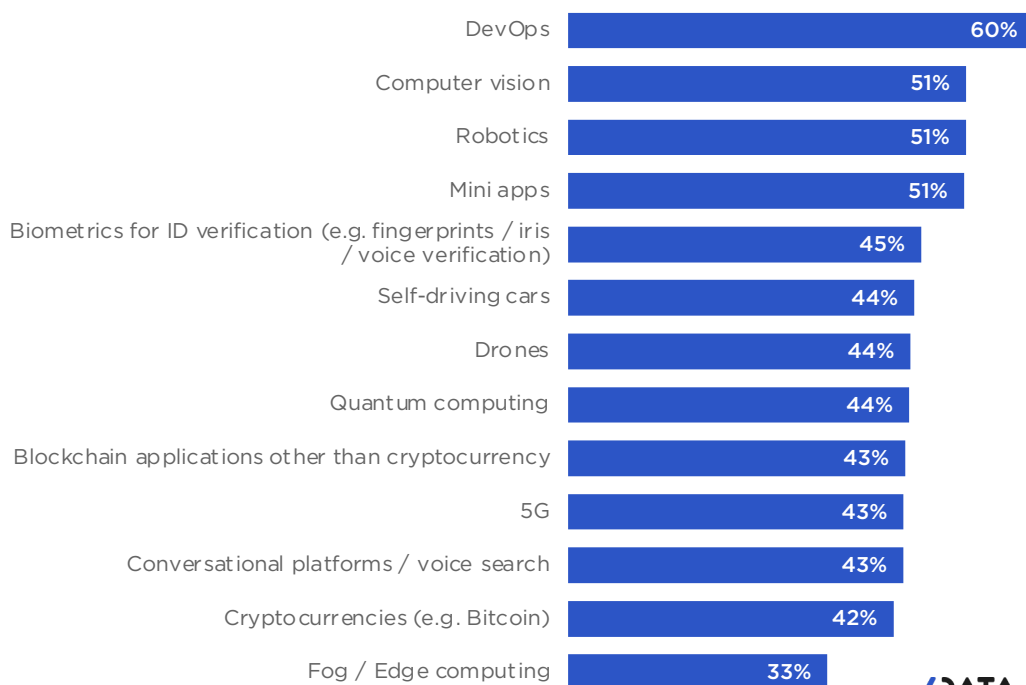
“ *DevOps has reached mainstream adoption* ”

With proven commercial applications and a strong pipeline of learners, DevOps sits firmly in the high engagement/high adoption quadrant, and with very little year-on-year changes in adoption and engagement rates, this technology appears to have stabilised.

Despite this stabilisation, DevOps is still evolving and changing. However, the nature of these changes indicates that DevOps is becoming well-established. For example, we're already beginning to see companies offering DevOps-as-a-Service, and DevOps is giving rise to spinoff concepts, such as DevSecOps. When a technology becomes available “as-a-service”, or spins off new sub-disciplines, this is an indication that it has reached maturity. It still seems that there is plenty of room for change and growth in DevOps, but can we continue to classify it as an emerging technology? Probably not.

Engagement continues to be high for DevOps

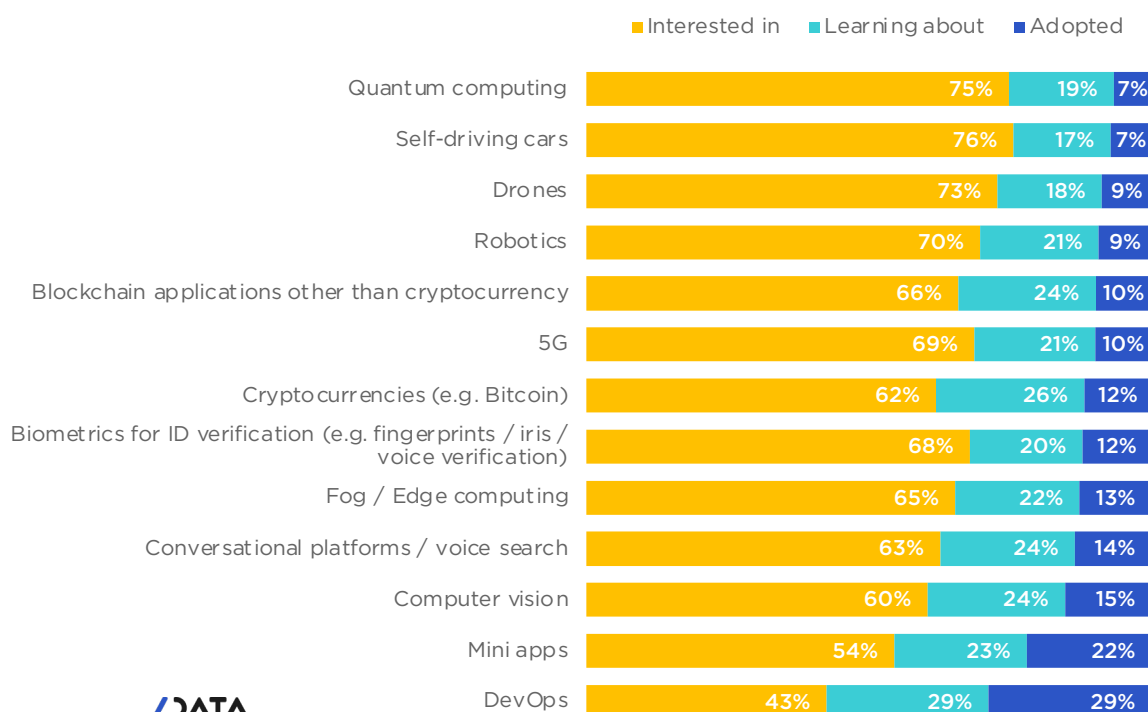
% of developers engaged with a technology (Q3 2020 n=15,876)



At the other end of the scale, we see that only a third of developers are engaged with fog/edge computing. As an emerging technology with primarily industrial applications, fog/edge computing doesn't have the wide-ranging commercial applications of DevOps to draw developers in, nor does it inspire imagination in the same way as robotics or quantum computing. This is why so few developers are engaged with it. For those that are engaged, however, more than a third are actively using the technology - either having adopted it at work or actively learning about it. To put this into context, 13% of engaged developers are currently working on this technology. This puts fog/edge computing in fifth place, approximately level with voice platforms (14%) and biometrics (12%). This means that fog/edge computing is a niche area in terms of engagement, but for those developers who do engage, it offers a great career growth opportunity. Due to these factors, fog/edge computing sits firmly in the low engagement/high adoption quadrant. That said, it is one of the only technologies where both engagement and adoption has increased year-on-year, the other being 5G. There was a 3 percentage points increase in the engagement rate, and a 6 percentage points increase in adoption - the largest increases we measured.

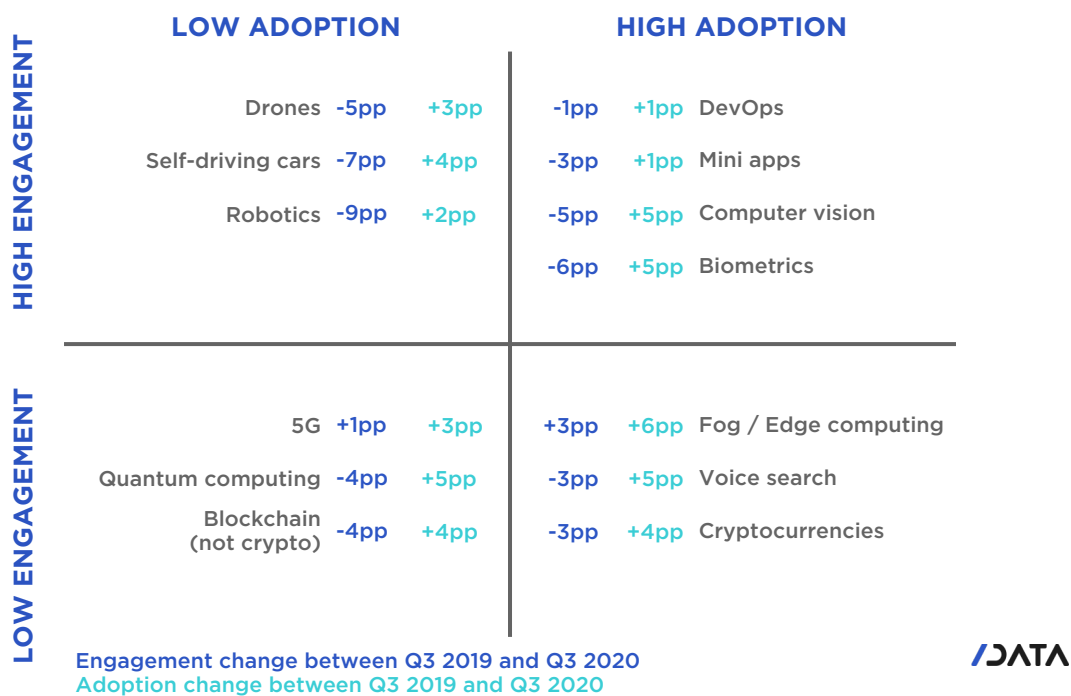
Large differences in adoption rates

% of engaged developers working on, learning about, or interested in a technology (Q3 2020 n=15,876)



Engagement with most emerging technologies fell, but adoption rose for many

Engagement and adoption classifications and year-on-year change
Q3 2019 (n=16,055) | Q3 2020 (n=15,876)



Elsewhere, around half of developers are engaged with computer vision, mini apps, and robotics, but when we take a deeper look into how developers engage with these technologies, differences begin to emerge. Computer vision and mini apps have some of the highest adoption rates, after DevOps. 22% of developers engaged with mini apps are working on this technology. For computer vision, this drops to 15%. Robotics, on the other hand, has an adoption rate of just 9%, reflecting the higher barriers to entry here. Robotics hardware is an additional cost in which many developers may be unwilling to invest. These three technologies all saw year-on-year drops in engagement, with robotics falling by 9 percentage points - the largest drop we observed - computer vision falling by 5 percentage points and mini apps falling by 3 percentage points. This said, the adoption rate for computer vision rose by 5pp. As one of the largest increases, this shows that computer vision is on the path to maturity and, with almost a quarter of engaged developers actively learning about this topic, we can expect this technology to continue to mature in this way.

“ A strong pipeline of learners indicates that computer vision is on the path to maturity

Around two-thirds of all the technologies we ask about have engagement rates between 42% and 45%. Despite being similar in this way, there exists substantial differences in how developers engage with this group of technologies. For example, there is a larger proportion of engaged developers who are learning about each blockchain application - cryptocurrencies (26%), and all other blockchain applications (24%) - than every other technology with a similar engagement rate (self-driving cars, drones, quantum computing, 5G, and voice platforms). These technologies have engagement rates between 18% and 21%. This shows that blockchain technologies have captured the interest of developers. Furthermore, with a 4 percentage points year-on-year increase in adoption, blockchain technologies are slowly becoming commercialised.

Finally, there are several technologies which don't stand out from the crowd in terms of their usage profile, but when we look at the year-on-year change in the engagement and adoption rates, an interesting story comes to the fore. Self-driving cars, robotics, drones, and quantum computing all had falling engagement rates when compared to last year, but for each of these technologies, the adoption rate rose by a small amount. It seems that many developers are becoming fatigued with these technologies, slowly losing interest as these promising technologies repeatedly fail to achieve mainstream adoption. But those developers who do engage continue to see some success, indicated by the rising adoption rates. Indeed, when controlling for the falling proportion of engaged developers, we still see a real, albeit small, rise in the absolute adoption rate. These technologies are coming, whether you like it or not.

Methodology

The Developer Economics Survey

Developer Economics 19th edition reached 17,000+ respondents from 159 countries around the world. As such, the Developer Economics series continues to be the most global independent research on mobile, desktop, industrial IoT, consumer electronics, third party app ecosystems, cloud, web, game, AR/VR, 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 June and August 2020.

Respondents to the online survey came from 159 countries, including major app and machine learning development hotspots such as the US, China, India, Israel, the UK, and Russia and 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 eight languages in addition to English - Simplified Chinese, Traditional Chinese, Spanish, Portuguese, Vietnamese, Russian, Japanese, and Korean - and promoted by more than 70 leading community and media partners within the software development industry.

Our respondents came from a broad age spectrum, from young coders who are under 18 to the seasoned ones over 55. As software development is still a man's world, 84% of our respondents were male and 14% female, excluding other options and those who did not specify their gender.

Respondents were asked which types of projects they are involved in out of the twelve 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, and apps/extensions for third party 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 weighted the regional distribution across eight regions by a factor that was determined by the regional distribution and growth trends identified in our Developer Economy research. Each of the separate branches: mobile, desktop, industrial IoT, consumer electronics, 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 modeling 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, 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 <https://www.slashdata.co/methodology>.

Join the global developer community!

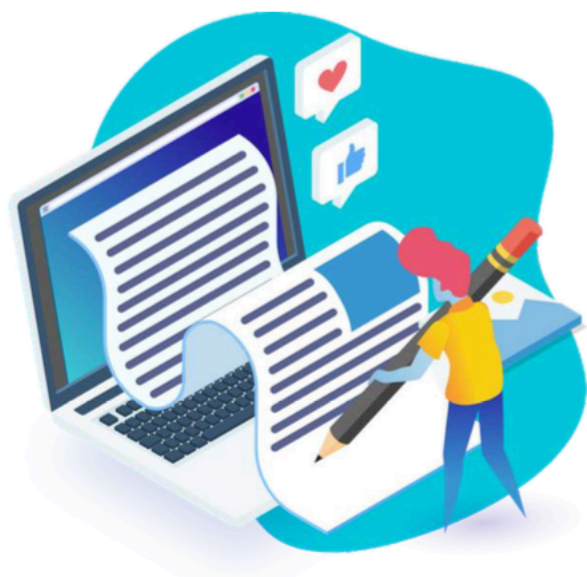
Developer Economics is more than just a survey.

It is a global community with 30,000+ developers from 160 countries. We bring together developers who care about what comes next in web, desktop, cloud, mobile, industrial IoT, consumer electronics, third party app ecosystems, AR/VR, data science and machine learning, and game development.

Become a part of our tribe:

- Shape the future of the developer ecosystem by taking part in our surveys twice a year.
- Win exclusive prizes and cash rewards via our loyalty program by participating in our research.
- Get free priority access to our research findings, reports, and graphs.
- Learn how data dictates new trends through our blog.
- Subscribe to our newsletter - get free resources, tech news, coding tips, and more twice a month, straight to your inbox. Plus, win awesome swag every month!

CLICK AND JOIN THE COMMUNITY!



Content creation partners

<https://www.developereconomics.com/blog>

Feel like you could provide value to our community with unique content? We want to hear your ideas!

Send your best blog post pitch to vanessa@slashdata.co



THE ANALYST OF THE DEVELOPER ECONOMY