STATE OF THE DEVELOPER NATION
20TH EDITION

The latest trends from our Q1 2021 survey of 19,000+ developers
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
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 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
SlashData © Copyright 2021 | Some rights reserved
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.

Jack Witkowski
Data Storyteller
Jack is a Data Storyteller with over five years of experience in data science and applied research. He holds an MSc in Investment Analysis and MPhil in Economics from Oxford University with a specialisation in econometrics. He has a passion for generating actionable insights and full stack development.

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.
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.

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.
ABOUT THIS REPORT

SlashData Developer Economics is the leading research programme on mobile, desktop, industrial IoT, consumer electronics, embedded, 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 20th Developer Economics global survey wave ran from November 2020 to February 2021 and reached more than 19,000 developers in 155 countries. This research report delves into key developer trends for Q1 2021 and beyond. The report focuses on six major themes - each with its own visualisations - showing how the data lends insight into the developer community.

- **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.

- **Machine learning developers and their data:** The data science (DS), machine learning (ML), and artificial intelligence (AI) field is adapting and expanding. From the ubiquity of data science in driving business insights to AI’s facial recognition and autonomous vehicles, data is fast becoming the currency of this century. This chapter focusses on this data and the profile of the developers who work with it.

- **On the influence of developers:** Developers are highly valued for their technical expertise, and are thus often considered to be key influencers when it comes to making technology decisions in companies. In this chapter, we show just how far their influence reaches and what factors leverage their power.

- **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 three years. In this chapter, we discuss which technologies have increased and decreased in popularity over the previous 12 months.

- **On COVID-19’s current and lasting effects:** The COVID-19 pandemic has fundamentally changed the way we work and learn. In this chapter, we provide insights into the extent to which developers shifted to remote working and learning during the pandemic. We also explore the distinct developer groups and regional communities that were affected the most by COVID-19.

- **Embedded software - An overview:** Embedded software offers many advantages in specific use cases - particularly where performance and power consumption are concerned. However, this type of development brings its own challenges, notably around ensuring maximum stability and security. For the first time in our global survey, we asked developers specifically about their experience in embedded software development. In this chapter, we offer a sneak preview of some of the data we collected. 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, Eve, Giannos, Jack, Jed, Konstantinos, Lazaros, Maria, Miljana, Moschoula, Natasa, Richard, Sam, Sarah, Sartios, Sofia, Stathis, Steve M., Steve V., Vanessa, and Viktorija at SlashData.
We’d like to thank everyone who helped us reach 19,000+ respondents for our survey and create this report. Our Media Partners - 4Scotty, DZone, HPE, Kontent, Microsoft, Samsung, and so many others.

A special thanks to the Meetups participating in our survey including: AI Festival Nigeria, Greece JS, R-Ladies Italy, and Social Hackers Academy.

Our Developer Committee supported our efforts once more, to create the most up-to-date and detailed survey. Special thanks to our members: Abdulrazaq, Amulya, Baldomero, Christopher, Dalton, Deborah, Dominic, Gilles, Karmal, Mark, and Mihai, 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

Programming language communities – An update
• JavaScript is the most popular programming language by some distance, with nearly 14M developers using it globally.
• Since it surpassed Java in popularity at the beginning of 2020, Python has remained the second most widely adopted language behind JavaScript, with just over 10M users.
• Google’s preferred language for Android development, Kotlin, has grown more than two-fold in size since the end 2017.
• Rust and Lua were the two fastest growing languages communities in the past year.

Machine learning developers and their data
• With 68% usage, unstructured text data is the most common type of data that machine learning (ML) developers and data scientists work with.
• The majority (53%) of ML developers and data scientists are professionals - although, in some cases, they might be also involved in hobby or student projects in parallel.
• 65% of those who are exclusively students use one or two types of data, while 61% of exclusively hobbyists and only 54% of exclusively professionals use one or two types. Very large training datasets aren’t perhaps as ubiquitous as one might expect.

On the influence of developers
• A substantial 75% of hands-on developers and 92% of developer team leads are, to some extent, involved in tool buying decisions.
• Developer team leads are particularly influential in their role. 63% of them make recommendations or influence their company’s decision makers.
• Highly experienced, hands-on developers without a lead role are deemed to be as influential in purchasing decisions as product managers.

Emerging technologies
• Adoption of mini apps is on the rise, as developers recognise their broad, practical applications.
• Blockchain applications attract developers on an academic level, and there remain many opportunities for commercial adoption.
• Brain / body computer interfaces sit in the sci-fi realm, along with quantum computing and self-driving cars.
• Low engagement and adoption of hearables, DNA computing / storage, and haptic feedback demonstrate the nascent nature of these technologies.

On COVID-19’s current and lasting effects
• 37% of developers say that the pandemic changed nothing in the way they work, and 43% in the way they study.
• Developers working for large companies were more likely to go fully remote during the pandemic.
• Junior developers were affected the least by the pandemic in the way they work.
• Switching to remote working has been more common in Western regions.
• Younger learners, most likely those who are studying for a formal degree, were the most affected in the way they learn.

Embedded software - An overview
• One in ten developers work on embedded software projects.
• Embedded developers are more likely to also be involved in industrial IoT, consumer electronics, augmented reality, and virtual reality than developers not involved in embedded software.
• Most embedded developers target desktop environments with their code, but these developers are less likely to also target other environments.
• C++ is the most popular programming language amongst embedded developers - it’s used by 40% of them.
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.

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, as of Q1 2021, there are 24.3 million 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 have consistently asked 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.

### Size of programming language communities in Q1 2021
Active software developers, globally, in millions (Q1 2021 n=13,348)

<table>
<thead>
<tr>
<th>Language</th>
<th>Size (in millions)</th>
<th>Most popular in</th>
<th>Least popular in</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript*</td>
<td>13.8 M</td>
<td>Web, Cloud</td>
<td>DS/ML, Embedded</td>
</tr>
<tr>
<td>Python</td>
<td>10.1 M</td>
<td>DS/ML, IoT apps</td>
<td>Mobile, Web</td>
</tr>
<tr>
<td>Java</td>
<td>9.4 M</td>
<td>Mobile, Cloud</td>
<td>DS/ML, Web</td>
</tr>
<tr>
<td>C/C++</td>
<td>7.3 M</td>
<td>Embedded, IoT apps</td>
<td>Web, Cloud, Mobile</td>
</tr>
<tr>
<td>C#</td>
<td>6.5 M</td>
<td>AR/VR, Desktop</td>
<td>DS/ML, Mobile</td>
</tr>
<tr>
<td>PHP</td>
<td>6.3 M</td>
<td>Web, Cloud</td>
<td>DS/ML, Mobile</td>
</tr>
<tr>
<td>Visual development tools</td>
<td>3.3 M</td>
<td>Desktop, AR/VR</td>
<td>Cloud, Web</td>
</tr>
<tr>
<td>Kotlin</td>
<td>2.6 M</td>
<td>Mobile, AR/VR</td>
<td>Cloud, IoT devices</td>
</tr>
<tr>
<td>Swift</td>
<td>2.5 M</td>
<td>Mobile, AR/VR</td>
<td>DS/ML, Desktop</td>
</tr>
<tr>
<td>Go</td>
<td>2.1 M</td>
<td>Cloud, AR/VR</td>
<td>DS/ML, Web</td>
</tr>
<tr>
<td>Ruby</td>
<td>1.8 M</td>
<td>IoT, Cloud</td>
<td>DS/ML, Web</td>
</tr>
<tr>
<td>Objective C</td>
<td>1.6 M</td>
<td>AR/VR, IoT apps, Embedded</td>
<td>Desktop, games</td>
</tr>
<tr>
<td>Rust</td>
<td>1.3 M</td>
<td>AR/VR, IoT apps</td>
<td>Web, Cloud, Desktop</td>
</tr>
<tr>
<td>Lua</td>
<td>1.0 M</td>
<td>AR/VR, IoT</td>
<td>Mobile, Desktop</td>
</tr>
</tbody>
</table>

(* JavaScript includes CoffeeScript and TypeScript)

(*) JavaScript includes CoffeeScript and TypeScript

Developer Economics | State of the Developer Nation Q1 2021 | © SlashData | Some rights reserved
JavaScript is the queen of programming languages

JavaScript is the most popular programming language by some distance, with nearly 14M developers using it globally. More importantly, the JavaScript community has been growing in size consistently for the past three years. Between Q4 2017 and Q1 2021, more than 4.5M developers joined the community - the highest growth in absolute terms across all languages. Even in software sectors where JavaScript is not among developers’ top choices, like data science or embedded development, about a fourth of developers use it in their projects.

Since it surpassed Java in popularity at the beginning of 2020, Python has remained the second most widely adopted language behind JavaScript. Python now counts just over 10M users, after adding 1.6M net new developers in the past year alone. That’s a 20% growth rate, the highest across all the large programming language communities of more than 6M users. The rise of data science and machine learning (ML) is a clear factor in Python’s popularity. Close to 70% of ML developers and data scientists report using Python. For perspective, only 17% use R, the other language often associated with data science.

Java is the cornerstone of the mobile app ecosystem - Android - as well as one of the most important general-purpose languages. Although it has been around for more than two decades now, its traction among developers keeps growing steadily. Since the end of 2017, nearly 2.5M developers have joined the Java community, which now counts 9.4M developers.

The group of major, well-established languages is completed with C/C++ (7.3M), C# (6.5M) and PHP (6.3M). C# lost three places in the rankings of language communities between Q3 2019 and Q3 2020, but it regained its lead over PHP in the past six months after adding half a million developers.

C and C++ are core languages in embedded and 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# is traditionally popular within the desktop developer community, but it’s also the most broadly used language among AR/VR and game developers, largely due to the widespread adoption of the Unity game engine in these areas.

Kotlin’s rise continues

The fastest growing language community in percentage terms is Kotlin. In fact, it’s one of the two communities - the other being Rust - that has grown more than two-fold over the last three years, from 1.1M developers in Q4 2017 to 2.6M in Q1 2021. This is also very evident from Kotlin’s ranking, where it moved from 11th to eight place during that period - a trend that’s largely attributed to Google’s decision to make Kotlin its preferred language for Android development. Even so, Kotlin still has a long way to go to catch up with the leading language in mobile development, Java; there are currently twice as many mobile developers building applications in Java than in Kotlin.

Swift was recently outranked by Kotlin, after attracting slightly fewer net new developers in the second half of 2020 (100K vs 300K). Even so, Swift is currently the default language for development across all Apple platforms, which has led to a stagnation in the adoption of Objective C. This gradual phase-out of Objective C from the Apple app ecosystem is also matched by a significant drop in its rank, from ninth to 12th place.

Rust and Lua were the two fastest growing language communities in the last 12 months

The more niche languages - Go, Ruby, Rust, and Lua - are still much smaller, with up to 2.1M active software developers each. Go and Ruby are important languages in backend development, but Go has grown slightly faster in the past year, both in absolute and percentage terms. Rust has formed a very strong community of developers who care about performance, memory safety, and security. As a result, it grew faster than any other language in the last 12 months, more than doubling in size. Finally, Lua was also among the fastest growing language communities in the last year, mainly attracting AR/VR and IoT developers looking for a scripting alternative to low-level languages such as C and C++.
Ranking of programming language communities, 2017-2021

(*) JavaScript includes CoffeeScript and TypeScript
The data science (DS), machine learning (ML), and artificial intelligence (AI) field is adapting and expanding. From the ubiquity of data science in driving business insights, to AI’s facial recognition and autonomous vehicles, data is fast becoming the currency of this century. Read on to learn more about this data and the profile of the developers who work with it.
It takes all types

The different types of ML/AI/DS data and their applications

We ask developers in ML, AI, and DS what types of data they work with. We distinguish between unstructured data - images, video, text, and audio - and structured tabular data. The latter group includes tabular data that they may simulate themselves.

With 68% of ML/AI/DS developers using unstructured text data, it is the most common type of data these developers work with; however, developers frequently work with multiple types of data. Audio is the most frequently combined data type: 75-76% of those that use audio data also use images, video, or text.

Given the most popular applications of audio data are text-to-speech generation (47%) and speech recognition (46%), the overlaps with video and text data are clear. Image data, like audio, overlaps heavily with video data: 78% of those using video data also use image data.

For the first time, we have tracked the application areas that ML/AI/DS developers are using their data in

The reverse dependence isn’t as strong: only 52% of those using image data are also video data users. The top two applications of both these data types are the same: image classification and facial recognition. These are two key application fields driving the next generation of intelligent devices: improving augmented reality in games and underpinning self-driving cars, in home robotics, home security surveillance, and medical imaging technology.
With 59% usage, tabular data is the second most popular type of data. 92% of the tabular data ML/DS/AI developers use is observed, while the other 8% is simulated. The two most common use-cases for this data is workforce planning - 39% of developers who use simulation do this - and resource allocation, also at 39%.

Structured tabular data is least likely to be combined with other types of data. Although uncommon to combine this type of data with audio or video data, 69% do combine tabular data with unstructured text data. The top application of both tabular data and unstructured text is the analysis and prediction of customer behaviour. This is the sort of analysis often done on the data nuggets we leave behind when searching on retail websites - these are key inputs to algorithms for natural language and recommender systems.
# Keeping it strictly professional?

## The professional status of ML/AI/DS developers

ML/AI/DS developers engage in their fields at different levels. Some are professionals, others students or hobbyists, and some are a combination of the above. The majority (53%) of all ML/DS/AI developers are professionals - although they might not be so exclusively.

Of all the data types, audio data has the highest proportion of professional ML/DS/AI developers. 64% of ML/AI/DS developers who use this type of data classified themselves as a professional; and the majority (50%) of these professionals are applying audio data to text-to-speech generation. The high proportion of professionals in this field might be a byproduct of co-influencing variables: audio data is the data type most frequently combined with other types, and professionals are more likely to engage with many different types of data.

Data types popular with students include image, tabular, and text data. Between 18-19% of developers who work with these types of data are students. There are many well-known datasets of these types of data freely available. With this data in hand, students also favour certain research areas.

### The professional / hobbyist / student mix in the ML/AI/DS ecosystem

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Exclusively professionals</th>
<th>Professionals and either hobbyists or students</th>
<th>Exclusively students</th>
<th>Hobbyists and students but not professionals</th>
<th>Exclusively hobbyists</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>43%</td>
<td>21%</td>
<td>13%</td>
<td>17%</td>
<td>7%</td>
</tr>
<tr>
<td>Video</td>
<td>42%</td>
<td>19%</td>
<td>13%</td>
<td>19%</td>
<td>6%</td>
</tr>
<tr>
<td>Tabular data (including simulated data)</td>
<td>36%</td>
<td>19%</td>
<td>18%</td>
<td>19%</td>
<td>8%</td>
</tr>
<tr>
<td>Text</td>
<td>35%</td>
<td>19%</td>
<td>18%</td>
<td>19%</td>
<td>9%</td>
</tr>
<tr>
<td>Images</td>
<td>33%</td>
<td>20%</td>
<td>19%</td>
<td>20%</td>
<td>8%</td>
</tr>
</tbody>
</table>

% of machine learning, data science, or artificial intelligence developers (Q1 2021 n=2,766)
Besides differences in application areas, students, hobbyists, and professionals engage with varying types of data. 65% of those who are exclusively students use one or two types of data, while 61% of exclusively hobbyists and only 54% of exclusively professionals use one or two types. Developers who are exclusively professionals are the most likely to be relying on many different types of data: 23% use four or five types of data. In contrast, 19% of exclusively hobbyists and 15% of exclusively students use four to five types. Level of experience, application, and availability of datasets all play a role in which types of data an ML/AI/DS developer uses. The size of these datasets is the topic of the next section.
Is all data ‘big’?

The size of ML/AI/DS developers’ structured and unstructured training data

The hype around big data has left many with the impression that all developers in ML/AI/DS work with extremely large datasets. We asked ML/AI/DS developers how large their structured and unstructured training datasets are. The size of structured tabular data is measured in rows, while the size of unstructured data - video, audio, text - is measured in disc size. Our research shows that very large datasets aren’t perhaps as ubiquitous as one might expect.

The most common size band is 1K - 20K rows of data, with 25% of ML/AI/DS developers using structured training datasets of this size. This differs by application type. For example, 22% of those working in simulation typically work with 20K - 50K rows of data; while 21% of those working with optimisation tools work with 50K - 100K rows of data.

Dataset size also varies by professional status. Only 11% of exclusively professional developers use structured training datasets with up to 20K rows, while 43% of exclusively hobbyists and 54% of exclusively students use these small datasets.

This may have to do with access to these datasets - many companies generate large quantities of data as a byproduct or direct result of their business processes, while students and hobbyists have access to smaller, open-source datasets or those collected via their learning institutions.

A further consideration is, who has access to the infrastructure capable of processing large datasets? For example, those who are exclusively students might not be able to afford the hardware to process large volumes of data.

Non-tabular data is a useful measure for comparisons within categories: for example, 18% of image datasets are between 50MB-500MB, while only 8% are more than 1TB in size. The measure doesn’t, however, allow for cross-type comparisons, since different types of data take up different amounts of space. For example, 50MB of video data takes up a considerably shorter length of time than 50MB of audio data.
The categorisation of the different data sizes was designed to take into account the steps in required processing power. For most ML/AI/DS developers, we expect that a 1-25GB dataset could be handled with powerful, but not specialised, hardware. Depending on the language and modelling method used, 25GB on disc relates to the approximate upper bound in memory size that this type of hardware could support.

We see that 26% of ML/AI/DS developers using text data and 41% using video data will require specialised hardware to manage their training. The high level of specialised hardware manifests as a barrier-to-entry: data analysis on these large datasets is beyond an achievable scope without the backing of deep pockets supplying cloud-based technology support or infrastructure purchases.
ON THE INFLUENCE OF DEVELOPERS

I once worked for a large digital company that was about to modernise its infrastructure by selecting a new cloud platform vendor. A small team of six developers was put in charge of arranging a proof-of-concept (POC) that would demonstrate the feasibility of this venture. After six weeks of getting our hands dirty, some developers started to voice their dislike for the new platform’s clunky interface. Suddenly, a multi-million dollar commitment was at the mercy of a few hands-on experts and proverbially hanging by a thread.
In a now-classic study undertaken in the early 1960s, John French and Bertram Raven discovered how different types of power affect one’s ability to lead an organisation. The one that first springs to mind is legitimate power and it comes from holding a position of authority within an organisation, such as being a team lead, associate director, or chief executive.

Yet, one does not always need to be sitting in a managerial chair to become a key influencer in a company. Expert power, for example, does not come with the rank but rather through one’s experiences, skills, or knowledge and it is as effective in shaping a company’s decisions. As tech employees gain expertise in particular areas, they become trusted members and natural thought leaders in those areas. Our data shows that 75% of hands-on developers and 92% of developer team leads are regularly involved in a company’s purchasing decision.

Executives responsible for managing the budget or pivoting their company’s strategy are frequently consulting seasoned developers before calling the big shots when purchasing new tools or agreeing on specifications.

Our survey has evaluated the voice of modern developers and their influence on key organisational processes such as making recommendations, approving budgets, or having a say in the final selection of purchasing company tools. For this chapter, we only consider professional developers in organisations and exclude hobbyists or students and those who work on their own, where a purchase decision is purely personal. We also compare our results across four organisational roles of developers: hands-on developers and developers in a multi-role, such as developer team leads, product managers, and CTOs who sit at the top of the developer food chain.
We find that 42% of hands-on developers and a substantial 64% of developers team leads are making recommendations or influencing their company’s decision makers. No other role appears as influential as a developer team lead - only solution architects coming a close second with 57%. Interestingly, developer team leads often step out of the shadows with 33% stating that they are making the final selection decision for purchasing company tools and 40% are handling specifications.

By uniting both legitimate and expert power, the role of a developer team lead becomes a highly influential one within an organisation, and thus sits between a CTO and a product manager. Even compared to CTOs, the role is ahead by a nose when it comes to giving recommendations; however, CTOs remain the commander-in-chief for making the final selection decision when purchasing (54%) or approving expenses (33%).
Interestingly, a substantial 42% of hands-on developers without a lead position state that they are making recommendations or influencing decision makers. We dug a little bit deeper to understand what leverages their claim and, indeed, our data has revealed that experience is a key factor for becoming more influential in any single category. While just 32% of developers with less than one year of experience state that they regularly sway decision-makers, over 50% of seasoned developers with six or more years of experience answered the same. Moreover, although just 6% of inexperienced developers are responsible for making the final selection decision for team/company tools, the figure is nearly doubled for developers with six years or more under their belts. We find that hands-on but highly experienced developers without a lead role are thus comparable to that of a product manager.
Moreover, our data shows that the strong correlation between a developer’s experience and their influence is not governed by company size. This can be further explained by taking into account that many modern tech companies and start-ups operate similarly in the way their developers are organised; in a flat hierarchy with small agile teams. For instance, many companies rely on implementing a POC to demonstrate that their idea could, in reality, be built or a new tool integrated. A few chosen developers with mixed experiences and seniority levels are on the frontline ploughing through test deployments. The verdict on the suitability of an idea is ultimately made by a small team of doers who collaborate and love to get their hands dirty. Although at the end they will present the outcome to their managers, stakeholders, or potential investors, it is often their intimate counsel and expertise in their field that will be the decisive factor.

Highly experienced, hands-on developers without a lead role are as influential in purchasing decisions as product managers.
As interest in a technology waxes and wanes, so does its influence, and few recent technologies have influenced software development practices as much as DevOps. This technology has become truly mainstream, seeing widespread adoption across software sectors, industries, and roles. We are delighted to say that, for these reasons, DevOps has matured out of our emerging technology tracker and instead has been replaced with several new and exciting technologies that have the potential to reshape the world. Here, we'll use developers’ engagement with and adoption of these technologies to help us understand just how this might come to pass.
We have tracked developers’ engagement with and adoption of different technologies over six surveys, spanning three years, ending Q1 2021. 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.
Developers are most engaged with robotics, mini apps, and computer vision

% of developers engaged with a technology
(Q1 2021 n=13,433)

- Robotics: 50%
- Mini apps: 50%
- Computer vision: 49%
- Self-driving cars: 45%
- Cryptocurrencies (e.g. Bitcoin): 45%
- Blockchain applications other than cryptocurrency: 44%
- Drones: 43%
- Biometrics for ID verification (e.g. fingerprints / iris / voice verification): 43%
- 5G: 43%
- Quantum computing: 43%
- Conversational platforms / voice search: 42%
- Brain / body computer interfaces (e.g. advanced prosthetics, neural lace): 40%
- DNA computing / storage: 34%
- Fog/edge computing: 31%
- Haptic feedback: 28%
- Hearables: 26%

After graduating DevOps from our emerging technology tracker, robotics, mini apps - apps embedded within another app - and computer vision head the table for those emerging technologies with which developers are most engaged. Around half of developers say they are working on, learning about, or interested in each of these technologies, and, whilst mini apps are most widely adopted by professional developers, hobbyists and students are most interested in robotics. However, of the developers engaged with mini apps, nearly a quarter are currently working on the technology. For computer vision, this drops to 15%, and for robotics, just 10%. Despite engaging developers in similar ways, it’s clear that the practical applications of mini apps are widely recognised by developers - in fact adoption increased by four percentage points in the last twelve months, one of the largest increases we saw.
Almost three in ten engaged developers are learning about cryptocurrencies, the most of any technology - though other blockchain applications are close behind on 26%. The academic interest in these technologies has yet to translate directly into adoption - only 14% and 12% of engaged developers are actively working on projects using these technologies. More than 40% of them are professionally involved in web apps / Software as a Service (SaaS), and a third are involved in mobile development as professionals. This said, adoption did increase for both cryptocurrencies (+5 percentage points), and other blockchain applications (+4 percentage points) in the last twelve months - developers are continuing to find practical applications for these technologies. With giants such as Maersk incorporating blockchain technology into their logistics management systems in the last few years, more widespread adoption is inevitable.

Quantum computing and self-driving cars continue to languish near the bottom in terms of adoption, but continue to spark some developers’ imaginations - more than two in five developers are engaged with these technologies. However, of these developers, fewer than one in ten are actually working on each of these technologies, and whilst engagement with these technologies dropped over the last twelve months, adoption increased for both - though more for quantum computing (4 percentage points) than self-driving cars (2 percentage points). There is a similar story with brain / body computer interfaces, which is a new technology that we added in the most recent survey - many developers are engaged, but, unsurprisingly, given its bleeding-edge status, very few are actively working on the technology.
More than a fifth of developers are learning about many emerging technologies

% of engaged developers working, learning about, or interested in a technology  (Q1 2021 n=13,433)

- Mini apps: Adopted 24%, Learning about 24%, Interested in 52%
- Computer vision: Adopted 15%, Learning about 25%, Interested in 59%
- Cryptocurrencies (e.g. Bitcoin): Adopted 14%, Learning about 29%, Interested in 57%
- Conversational platforms / voice search: Adopted 14%, Learning about 25%, Interested in 61%
- Biometrics for ID verification (e.g. fingerprints / iris / voice verification): Adopted 17%, Learning about 22%, Interested in 65%
- Fog/edge computing: Adopted 13%, Learning about 24%, Interested in 63%
- Blockchain applications other than cryptocurrency: Adopted 12%, Learning about 26%, Interested in 62%
- Haptic feedback: Adopted 11%, Learning about 22%, Interested in 67%
- 5G: Adopted 11%, Learning about 25%, Interested in 63%
- DNA computing / storage: Adopted 10%, Learning about 21%, Interested in 69%
- Hearables: Adopted 10%, Learning about 22%, Interested in 68%
- Robotics: Adopted 10%, Learning about 22%, Interested in 68%
- Drones: Adopted 9%, Learning about 20%, Interested in 71%
- Brain / body computer interfaces (e.g. advanced prosthetics, neural lace): Adopted 8%, Learning about 19%, Interested in 72%
- Quantum computing: Adopted 8%, Learning about 21%, Interested in 72%
- Self-driving cars: Adopted 7%, Learning about 18%, Interested in 75%
We also recently added hearables, DNA computing / storage, and haptic feedback to our list of emerging technologies. Engagement is low with these technologies; on a level with fog/edge computing - between a quarter and a third of developers are engaged. We see that around one in ten engaged developers are actively working on these very nascent technologies, and two in ten are learning about them. Though the engaged audience for these technologies is small, there is a core of developers contributing to their continued progress.

For each of the emerging technologies we have discussed, there are different barriers to widespread adoption. For many, the barriers are technological - the advances needed to bring quantum or DNA computing to the mainstream are many years away, but there are also social, cultural, and even legislative barriers which will impede progress. Though important, developers are only part of the puzzle.
### Blockchain applications are coming closer to the mainstream

Interest and adoption change from one year ago (Q1 2020 n=16,068 | Q1 2021 n=13,433)

<table>
<thead>
<tr>
<th>Technology</th>
<th>Low Adoption</th>
<th>High Adoption</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High Interest</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-driving cars</td>
<td>-1%</td>
<td>2%</td>
</tr>
<tr>
<td>Drones</td>
<td>-1%</td>
<td>3%</td>
</tr>
<tr>
<td>Robotics</td>
<td>-4%</td>
<td>2%</td>
</tr>
<tr>
<td>5G</td>
<td>1%</td>
<td>4%</td>
</tr>
<tr>
<td>Quantum computing</td>
<td>-3%</td>
<td>4%</td>
</tr>
<tr>
<td>Brain / body computer interfaces</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Hearables</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>DNA computing / storage</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Low Interest</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cryptocurrencies</td>
<td>4%</td>
<td>5%</td>
</tr>
<tr>
<td>Mini apps</td>
<td>0%</td>
<td>4%</td>
</tr>
<tr>
<td>Blockchain applications</td>
<td>-1%</td>
<td>4%</td>
</tr>
<tr>
<td>other than cryptocurrency</td>
<td>-2%</td>
<td>-2%</td>
</tr>
<tr>
<td>Biometrics for ID verification</td>
<td>-4%</td>
<td>3%</td>
</tr>
<tr>
<td>Fog/edge computing</td>
<td>-1%</td>
<td>4%</td>
</tr>
<tr>
<td>Conversational platforms</td>
<td>-2%</td>
<td>4%</td>
</tr>
<tr>
<td>/ voice search</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Haptic feedback</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**ENGAGEMENT CHANGE BETWEEN Q1 2020 AND Q1 2021**

**ADOPTION CHANGE BETWEEN Q1 2020 AND Q1 2021**
The COVID-19 pandemic has fundamentally changed the way people work and learn across industries, and developers are no exception to that. Although in the grand scheme of things, our data indicates that many developers have been weathering well the repercussions of an unprecedented crisis, there is much more to tell. In this chapter, we will deep dive into our survey data to find which developer groups and regional communities were affected the most by the pandemic and in what ways.
In our survey, we asked developers to assess the impact of the COVID-19 pandemic on the way they work or learn. In perhaps the most salient category, 7% of developers said that they had lost their job in the aftermath of the pandemic and 9% had dropped out of their studies. Notwithstanding the severity of becoming unemployed in times of crisis, the IT sector is still seen as one of the least impacted sectors in terms of hiring during the global pandemic, with an almost unwavering demand for professionals in software and hardware segments.

### How the COVID-19 pandemic affected the way developers work or study

<table>
<thead>
<tr>
<th>% of developers</th>
<th>How I work</th>
<th>How I study</th>
</tr>
</thead>
<tbody>
<tr>
<td>No effect, I’ve always been remote</td>
<td>25%</td>
<td>32%</td>
</tr>
<tr>
<td>No effect, I was never remote</td>
<td>12%</td>
<td>11%</td>
</tr>
<tr>
<td>Some effect, I went partially remote but I don’t expect this to last</td>
<td>15%</td>
<td>14%</td>
</tr>
<tr>
<td>Some effect, I went partially remote and I expect this to last</td>
<td>11%</td>
<td>10%</td>
</tr>
<tr>
<td>Significant effect, I went fully remote but I don’t expect to be forever</td>
<td>19%</td>
<td>16%</td>
</tr>
<tr>
<td>Significant effect, I went fully remote and I expect to be forever</td>
<td>11%</td>
<td>9%</td>
</tr>
<tr>
<td>Lost/quit job or dropped out of studies</td>
<td>7%</td>
<td>9%</td>
</tr>
</tbody>
</table>

Just 11% of all developers expect that they will keep working remotely forever.
Perhaps the most apparent impact of COVID-19 on the way developers work or learn has been a shift to partially or fully remote settings. Even so, 37% of developers say that the pandemic changed nothing about the way they work, and 43% for the way they study. Although this means that more than half of all developers have been affected in some way, and went either partly or fully remote, only 11% believe that they will keep being fully remote and 15% that they will keep being partly remote, forever. In support of this, many companies’ internal surveys have already spotted the first warning signs of their employees’ deteriorating mental health. The post-pandemic world might eventually have to allow for compromises.

Junior developers were affected the least by the pandemic in the way they work

% of developers in each experience level (Q1 2021 n=12,388)
Which developers were impacted the most?

Arguably, not every developer has been affected by their company’s decision to enforce remote working to the same extent. For example, does a working parent who juggles the daily demands of commutes and childcare embrace the opportunity of working remotely as a means of having a better work-life balance? On the other hand, if asked, how many pre-pandemic graduates would have, at the time of their study, agreed to go fully remote and potentially miss out on exploring the rich social life that a university offers to young people? Let’s take a closer look at some of our distinct developer groups to understand which factors have been having the greatest impact on ways of working and learning.

When looking at how the pandemic affected developers of different experience levels, we find that the more experienced developers were also more affected in the way they work. For instance, 40% of developers with less than one year of work experience say they were unaffected in their ways of working, compared to 35% of developers with six or more years of experience. While the gap is not particularly large, junior developers appear to have switched to remote working to a lesser extent. This may be, in part, due to younger people and new hires wanting to go to the office, get to know their colleagues, and connect with their peers.
Next, we evaluated COVID-19’s impact with respect to company size. We find that developers working for larger companies were clearly more affected in their ways of working by the pandemic. While 42% of developers in small companies between two and 50 employees say they were not affected, the number plummets to less than 30% for developers in companies of over 50 employees.

Large enterprises with more than 5,000 employees have been battling the repercussions of the pandemic at the frontlines; 51% of developers here went fully remote compared to just 29% of developers in companies with between two and 50 employees.
Note that, except for small companies, switching to fully-remote working was the most likely outcome in our survey. There are good reasons for this: large business organisations are naturally more risk-averse and commonly need large contiguous office spaces that have to be fully closed for all of their employees to effectively contain the spread of the virus. On the other hand, many small companies had more remote-friendly organisational structures to begin with. In particular, start-ups have been known to promote a remote-first culture due to the apparent benefits of lower seed capital and broader options to recruit and pool talents together.

Developers in small companies were less affected by the pandemic
Switching to remote working has been more common in Western regions

% of developers per region (Q1 2021 n=12,394 for all regions)

### West
- Western Europe & Israel: 33% Not affected by COVID-19, 25% Went partially remote due to COVID-19, 41% Went fully remote due to COVID-19
- South America: 36% Not affected by COVID-19, 23% Went partially remote due to COVID-19, 41% Went fully remote due to COVID-19
- North America: 40% Not affected by COVID-19, 26% Went partially remote due to COVID-19, 35% Went fully remote due to COVID-19
- Oceania: 29% Not affected by COVID-19, 37% Went partially remote due to COVID-19, 34% Went fully remote due to COVID-19

### Eastern Europe, Russia, Asia & Africa
- Eastern Europe, Russia & Former CIS: 44% Not affected by COVID-19, 24% Went partially remote due to COVID-19, 32% Went fully remote due to COVID-19
- South Asia: 38% Not affected by COVID-19, 31% Went partially remote due to COVID-19, 30% Went fully remote due to COVID-19
- Middle East & Africa: 45% Not affected by COVID-19, 28% Went partially remote due to COVID-19, 27% Went fully remote due to COVID-19
- East Asia: 44% Not affected by COVID-19, 36% Went partially remote due to COVID-19, 20% Went fully remote due to COVID-19

Perhaps it comes as no surprise when looking at different regions to find a substantial gap between the East and the West. Our data shows that the Western regions, such as Western Europe and the Americas, have more readily facilitated remote working for their employees. For instance, 41% of Western European developers went fully remote, as opposed to only 20% in East Asia. This could be due to a combination of different factors. For example, **pundits** argue that many Asian countries score low for having the technological infrastructure deemed necessary to adapt to remote working conditions, such as having poor home-office equipment or internet connectivity that is sensitive to traffic surges. Yes, social factors may have partly played a role, as **higher average household sizes and smaller apartments in emerging regions** pose roadblocks to their own for employees to balance their work and home life.
Lastly, we looked at COVID-19’s impact on learners. 39% of 18- to 24-year olds stated that the way they study has not been affected during the pandemic. On the contrary, among those students aged 25 and over, 50% or more were not affected. Thus, an interesting trend emerges here; that especially younger learners had to adapt by becoming partly or fully remote.

Our data offers one possible clue for this; younger learners are more likely enrolled in a formal degree programme than older learners, who are more likely to be self-taught and are to be found burning the midnight oil with online courses and boot camps that have traditionally fostered remote ways of studying.
Embedded software is vital in many situations - particularly where the environment predicates that application software cannot be used. Embedded development brings its own challenges, notably around working with fixed hardware requirements. For the first time, we asked developers specifically about their experience with embedded software development. Here’s a sneak preview of some of the data we collected.
In which other software sectors are embedded developers involved?

One in ten developers are involved in embedded software. However, developers are often involved in multiple software sectors. Amongst embedded software developers, web apps / Software as a Service (SaaS) is the most popular other sector, but it is also the most popular software sector amongst developers not involved in embedded software, indicating that there are no special synergies between the two sectors.

On the other hand, desktop is the second-most popular software sector for embedded developers, but only the fourth for those not involved in embedded development. This demonstrates the huge need for embedded software on desktop machines, in the form of drivers and other firmware. There may also be many skills that are transferable between these two sectors, for example, the use of C and C++ is common in both areas.

Furthermore, we see that embedded developers are more likely to also be involved in industrial IoT (IIoT) than their non-embedded counterparts - there are clearly strong synergies between these two technologies. We see a similar pattern for consumer electronics (CE), augmented reality (AR), and virtual reality (VR).

The disproportionately high involvement of embedded developers in these other software sectors is likely driven by very different circumstances - for these technologies, the hardware, rather than the software platform, is central to the development process. Some differences may still remain, however. For IIoT and CE, embedded development underpins the user experience - with few opportunities for direct user input, the hardware interactions become especially important. For AR and VR, however, the synergy with embedded development may be augmented by a need for performance. AR and VR headsets require enormous amounts of computing power in order to deliver a seamless user experience, and any performance gains are especially valuable here.
Where does embedded software developers’ code run?

The link between embedded software and desktop development becomes even more clear when we look at where embedded developers’ code runs. Nearly three in five embedded developers write code which runs on desktop or laptop computers, nearly double that of the next most popular environment; smartphones and tablets.

We also see that embedded development for desktop is something of a closed ecosystem - embedded developers who write code for desktop environments are less likely to also target most other environments, with the exception being smartphones and tablets - around 20% do so, compared to an average of approximately 30%.

Though important, the industrial applications of embedded software are less mainstream - a quarter of embedded developers write code for industrial computing units, and a fifth do so for network infrastructure. Nonetheless, these environments are still an important part of embedded software development.
Which languages do embedded developers use?

Embedded developers use a variety of languages for their embedded software projects, but C++ is by far the most popular; two in five embedded developers use this language, with C and Python close behind.

Further down the list, low-level languages such as Embedded C (16%), Arduino (14%), and Assembly (12%) are generally less popular, although their popularity changes greatly depending on the type of environment the developer is writing code for. For example, around a quarter of developers targeting CE devices other than smartphones, or non-x86 hardware architectures, are using Embedded C. Arduino is also more popular amongst these developers, as well as amongst those writing code for industrial computing units or network infrastructure.

We also see that high-level, interpreted languages - Python, Java, and JavaScript - are popular amongst embedded developers for their projects. This indicates that many embedded developers aren’t very close to the hardware; instead abstracting away some of the complexity through the use of operating systems. Indeed, with almost half of embedded developers using Android, the popularity of interpreted languages is clear.

<table>
<thead>
<tr>
<th>Programming language of choice for embedded development</th>
<th>% of embedded developers Q1 2021 (n=1,467)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++</td>
<td>40%</td>
</tr>
<tr>
<td>C</td>
<td>32%</td>
</tr>
<tr>
<td>Python</td>
<td>30%</td>
</tr>
<tr>
<td>Java</td>
<td>28%</td>
</tr>
<tr>
<td>JavaScript (incl. TypeScript, CoffeeScript)</td>
<td>25%</td>
</tr>
<tr>
<td>C#</td>
<td>19%</td>
</tr>
<tr>
<td>PHP</td>
<td>16%</td>
</tr>
<tr>
<td>Embedded C</td>
<td>16%</td>
</tr>
<tr>
<td>Arduino</td>
<td>14%</td>
</tr>
<tr>
<td>Assembly</td>
<td>12%</td>
</tr>
</tbody>
</table>
METHODOLOGY

The Developer Economics Survey

Developer Economics 20th edition reached 19,000+ respondents from 155 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, embedded, 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 November 2020 and February 2021.

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, 79% of our respondents were male and 20% 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 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 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, 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 https://www.slashdata.co/methodology.
Got a community of developers? Let’s partner up!

Spread the word about our surveys to the developers in your community and help their voices be heard by the tech industry! In return, we will help you:

- **Reach new developer audiences**
  Get your brand, events, tools and activities in front of our own community of 30,000+ developers globally, through logo placement, newsletter inclusions and content amplification on our blog.

- **Get valuable data & insights and make data driven decisions**
  Segment, understand and speak to your developer network using 25+ key graphs. Compare key trends in your developer community to the global average!

**LET’S JOIN FORCES!**

Contact us partnerships@slashdata.co