STATE OF THE DEVELOPER NATION

22ND EDITION

The latest trends from our Q1 2022 survey of 20,000+ 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 Nation

Developer Nation 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.developernation.net
@devnationworld
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 2022 | Some rights reserved
**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**  
*Director of Research*  
Konstantinos heads the Research Product team at SlashData and is responsible for all syndicated research products and custom research projects. With more than 10 years of experience as an engineer, consultant and manager, he oversees research planning, survey design, data analysis, insights generation and research operations.
ABOUT THE AUTHORS

Simon Jones  
*Data Storyteller*  
Simon has more than ten years’ experience working in research and software development across academic, industrial, and charitable sectors. He holds a PhD in Computer Science and has published over 40 peer-reviewed articles spanning areas such as mobile and ubiquitous computing, IoT, AR/VR, data science, and human-computer interaction.

Michael Condon  
*Data Storyteller*  
Michael is a Data Storyteller with over 15 years of applying data analysis across a range of roles and sectors. With experience in both the U.S and Europe, he’s used data to investigate operational loss, design innovation programs, and provide commercial insights to business leaders. He is responsible for interpreting data from SlashData’s developer surveys and discovering meaningful insights.
TABLE OF CONTENTS

ABOUT THIS REPORT

1. LANGUAGE COMMUNITIES - AN UPDATE

PARTNERS

2. UNDERSTANDING DEVELOPER PERSONALITIES

3. WHO IS USING LOW-CODE/NO-CODE TOOLS?

KEY INSIGHTS

4. SPOTLIGHT ON CHINA AND THE REST OF EAST ASIA

5. HOW DEVELOPERS GENERATE REVENUES

METHODOLOGY

6. EMERGING TECHNOLOGIES

TABLE OF CONTENTS
About This Report

SlashData’s Developer Nation survey 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 22nd edition of the Developer Nation global survey ran from December 2021 to February 2022 and reached more than 20,000 developers in 166 countries. This research report delves into key developer trends for Q1 2022 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.

- **Understanding developer personalities:** Personality theories provide a blueprint for understanding why people behave the way they do. In the latest edition of our Developer Nation survey, we incorporated a measure of the widely accepted ‘Big Five’ personality dimensions in order to better understand the personality traits of software developers. In this chapter, we segment the developer population into a set of distinct personality types and we discuss some key differences between these segments.

- **Who is using low-code/no-code tools?:** Low-code/no-code (LCNC) tools provide a visual approach to software development, abstracting and automating parts of the application development process. In this chapter, we investigate the extent to which developers are using LCNC tools, showing differences according to professional status, geographical regions, and experience levels.

1 Unless stated otherwise throughout the report, by “developers” we mean everyone who self-identifies as being involved in the software development ecosystem, including individuals holding business (e.g. product managers, C-level executives) and creative (e.g. UI/UX designers) roles.
• **Spotlight on China and the rest of East Asia:** This chapter focuses on some of the key differences between developers located in East Asia, including the Greater China region, and the rest of the world. Understanding these differences provides valuable insights that can help shape the strategy for developer engagement programmes.

• **How developers generate revenues:** How businesses and developers as individuals make money from software projects is one of the most important decisions they have to make. Of all the business models and strategies available, companies and freelancers need to pick the ones that best match their market and goals. In this chapter, we explore the revenue models used by professional developers and how these vary across industry verticals and companies of different sizes.

• **Emerging technologies:** Many high-profile emerging technologies are slated to change the world, including self-driving cars and the Metaverse. However, there are many more outside of the spotlight, such as AI-assisted programming, that will also have profound and far-reaching implications. In the last chapter, we explore how developers are involved with a range of emerging technologies.

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 Stathis Georgakopoulos, Product Marketing Manager for SlashData at stathis@slashdata.co. You can download and access this report and more free resources at [https://www.slashdata.co/free-resources/](https://www.slashdata.co/free-resources/)

You can find more developer data at [SlashData Dashboards](https://www.slashdata.co/). You can use filters to sort through the data by region, professional experience and more! Try it out here: [https://sdata.me/SDashboards](https://sdata.me/SDashboards)

Ahmed, Alex, Anastasia, Andreas, Christina, David, Eleni-Christina, Eleonora, Eve, Flora, Giannos, Jed, Konstantinos, Lampis, Lazaros, Maria, Michael C., Moschoula, Natasa, Natela, Nikita, Richard, Sam, Sarah, Simon, Sofia, Stathis, Steve M., Steve V., Valentin, Vanessa, and Viktorija at SlashData
Our Developer Committee supported our efforts once more, to create the most up-to-date and detailed survey. Special thanks to our members: Amulya, Cristian, Deborah, Dominic, Eduardo, Evan, Ioannis, Ivan, Jacob, James, Jeff, Joseph, JS, Kamil, Kujtim, Maxime, Mihály, Mikhail, Miroslav, Mustapha, Niccolò, Rajasekar, Rakesh, Ricardo, Sylv, and Thiago 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, Korean, Portuguese, Russian, Spanish, and Vietnamese. In addition, our prize payments partner, Chimoney.
KEY INSIGHTS

Language communities – An update
• JavaScript remains the most popular programming language for the tenth survey in a row, with close to 17.5M developers worldwide using it.
• Since surpassing Java in popularity two years ago, Python has remained the second most widely adopted language behind JavaScript. Python now counts 15.7M users.
• Rust has nearly tripled in size in the past 24 months, from just 0.6M developers in Q1 2020 to 2.2M in Q1 2022.
• Go and Ruby are important languages in backend development but Go has grown more than twice as fast in the past year in absolute terms.

Understanding developer personalities
• We used the Ten-Item Personality Inventory (TIPI) methodology in order to measure the ‘Big Five’ personality dimensions: emotional stability, extraversion, openness to experiences, agreeableness, and conscientiousness.
• We show five different developer personality profiles, across which we see associations with certain developer roles and types of information and content that developers prefer.
• High-conscientiousness and high-openness personality types use the widest range of information sources overall. Those who score highly for agreeableness are considerably more likely to turn to social media for information about software development.

Who is using low-code/no-code tools?
• 46% of professional developers use low-code/no-code (LCNC) tools for some portion of their development work.
• Experienced developers, particularly those with more than ten years of experience, are the least likely to use LCNC tools.
• Most developers that use LCNC tools do so for less than a quarter of their development work.
• The Greater China area has the highest LCNC tool adoption rate. 69% of developers in this region report using LCNC tools, compared to the global average of 46%.
• 19% of developers in North America use LCNC products for more than half of their coding work – almost twice the global average of 10%. This provides strong evidence that these tools can supplant traditional development approaches.
KEY INSIGHTS

Spotlight on China and the rest of East Asia

• The Greater China area has a relatively low concentration of highly-experienced developers (16+ years of development) when compared to developers in the rest of East Asia and the rest of the world.
• More than half of Chinese developers have learned how to code via undergraduate degrees in computing, which is about 10 percentage points more than developers in the rest of East Asia and the rest of the world.
• More than a quarter of developers in Greater China (26%) and the rest of East Asia (27%) don’t use Stack Overflow, which is more than three times the rate of developers in the rest of the world (8%).

How developers generate revenues

• Contracted development is the revenue model of choice across all industry verticals, used by nearly a third (31%) of professional developers.
• Less than one in ten (7%) professional developers are generating revenue from selling data.
• Usage of the advertising revenue model declines as companies grow in size.
• Developers working for large enterprises (5K+ employees) tend to use multiple revenue models less often than developers in smaller companies.

Emerging technologies

• AI-assisted development is especially salient to developers - engagement with this technology is the highest of any, and adoption is in the top three.
• Blockchain technologies, including cryptocurrencies and NFTs, have high learning and adoption rates amongst developers engaged with them.
• Interest in blockchain applications and cryptocurrencies has increased the most of any technology, but adoption of blockchain applications has stagnated.
• Growth in adoption rates in general has stagnated, but developers are instead widening their interests.
• 28% of developers engaged with the Metaverse are actively learning about it.
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 State of the 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 is 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 2022, there are 31.1 million active software developers in the world.

Second is 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 2022

<table>
<thead>
<tr>
<th>Language</th>
<th>Active Developers (Millions)</th>
<th>Most popular in</th>
<th>Least popular in</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript*</td>
<td>17.4 M</td>
<td>Apps for 3rd-party ecosystems, Web</td>
<td>DS/ML, Embedded</td>
</tr>
<tr>
<td>Python</td>
<td>15.7 M</td>
<td>DS/ML, IoT apps</td>
<td>Mobile, Web</td>
</tr>
<tr>
<td>Java</td>
<td>14.0 M</td>
<td>Cloud, Mobile</td>
<td>DS/ML, Web</td>
</tr>
<tr>
<td>C/C++</td>
<td>11.0 M</td>
<td>Embedded, IoT apps</td>
<td>Web, Cloud</td>
</tr>
<tr>
<td>C#</td>
<td>10.0 M</td>
<td>Desktop, Games</td>
<td>DS/ML, Mobile</td>
</tr>
<tr>
<td>PHP</td>
<td>7.9 M</td>
<td>Web, Cloud</td>
<td>DS/ML, Mobile</td>
</tr>
<tr>
<td>Kotlin</td>
<td>5.0 M</td>
<td>Mobile, AR/VR</td>
<td>DS/ML, Desktop</td>
</tr>
<tr>
<td>Swift</td>
<td>3.5 M</td>
<td>Mobile, AR/VR</td>
<td>Cloud, Embedded</td>
</tr>
<tr>
<td>Go</td>
<td>3.3 M</td>
<td>Cloud, Apps for 3rd-party ecosystems</td>
<td>Mobile, DS/ML</td>
</tr>
<tr>
<td>Objective C</td>
<td>2.4 M</td>
<td>AR/VR, IoT devices</td>
<td>Desktop, Apps for 3rd-party ecosystems</td>
</tr>
<tr>
<td>Rust</td>
<td>2.2 M</td>
<td>AR/VR, IoT apps</td>
<td>Web, Mobile</td>
</tr>
<tr>
<td>Ruby</td>
<td>2.1 M</td>
<td>IoT, Apps for 3rd-party ecosystems</td>
<td>Web, Embedded</td>
</tr>
<tr>
<td>Dart</td>
<td>1.8 M</td>
<td>Mobile, Apps for 3rd-party ecosystems</td>
<td>Web, Apps for 3rd-party ecosystems</td>
</tr>
<tr>
<td>Lua</td>
<td>1.4 M</td>
<td>Games, IoT devices</td>
<td>Mobile, Embedded</td>
</tr>
</tbody>
</table>

*JavaScript includes CoffeeScript and TypeScript
JavaScript remains the largest language community

JavaScript remains the most popular programming language for the tenth survey in a row, with close to 17.5M developers worldwide using it. Notably, the JavaScript community has been growing in size consistently for the past several years. Between Q1 2020 and Q1 2022, 5.2M developers joined the community - one of the highest growths in absolute terms across languages. Not only do new developers see it as an attractive entry-level language, but existing ones are also adding it to their skillset. Even in software sectors where JavaScript is not among developers' top choices, like in data science or embedded development, more than a fifth of developers use it in their projects.

Want to see a regional breakdown of this data? Go to SlashData Dashboards and apply the "region" filter to see where developers live: https://sdata.me/SDashboards

Since surpassing Java in popularity two years ago, Python has remained the second most widely adopted language behind JavaScript, with the gap between the two largest communities gradually closing. Python now counts 15.7M users after adding 3.3M net new developers in the past six months alone. For context, JavaScript added just over 1M developers during the same period. The rise of data science and machine learning (ML) is a clear factor in Python’s growing popularity. About 70% of ML developers and data scientists report using Python. In comparison, only 17% use R, the other language often associated with data science.
Java is one of the most important general-purpose languages, as well as the cornerstone of the Android app ecosystem. Although it has been around for more than two decades now, it keeps experiencing strong and steady growth. Nearly 5M developers have joined the Java community since the beginning of 2021. Our data suggest that Java's growth is fueled not only by the usual suspects, i.e. backend and mobile development, but also by its rising adoption in AR/VR projects.

The group of major, well-established languages is completed with C/C++ (11M), C# (10M), and PHP (7.9M). Of these, PHP has grown the least in the past six months, with an increase of 0.6M net new developers between Q3 2021 and Q1 2022. As a result, it remains behind C#, which added nearly four times as many developers during the same period (2.2M).

C and C++ are core languages in embedded and IoT projects for both on-device and application-level coding, but they are also quite important in desktop development - a sector that attracts over 15M developers as of Q1 2022. PHP may still be the second most commonly used language in web applications after JavaScript, but C# is traditionally popular in many different areas of software development, including desktop and game development.

Rust’s growth continues

In the previous edition of this report, we had identified Rust as the fastest growing language community. This continues to be the case as of the beginning of 2022, with Rust having nearly tripled in size in the past 24 months, from just 0.6M developers in Q1 2020 to 2.2M in Q1 2022. It also moved from 14th to 12th place in the ranking of programming languages in just six months. Rust has formed a very strong community of developers who care about performance, memory safety, and security. As a result, it is mostly used in IoT software projects but also in AR/VR development, most commonly for implementing the low-level core logic of AR/VR applications.
Swift counts 3.5M developers and is currently the default language for development across all Apple platforms, which has led to a gradual phase-out of Objective C (2.4M) from the Apple app ecosystem. Nonetheless, Objective C has seen a slight uptick in its adoption for on-device IoT and AR/VR coding, which resulted in a small increase in its rank, from 12th to 11th place.

The more niche languages - Go, Ruby, Dart, and Lua - are still much smaller, with up to 3.3M active software developers each. Go and Ruby are important languages in backend development but Go has grown more than twice as fast in the past year in absolute terms, likely due to the fast development cycle it offers even though it is a compiled language. Dart has seen steady growth in the last year, propelled predominantly by the increasing adoption of the Flutter framework in mobile development. Finally, Lua was the third fastest growing language community in the last year, behind Rust and Kotlin, mainly attracting game and IoT developers looking for a scripting alternative to low-level languages such as C and C++.

Kotlin is another language we have consistently singled out for its impressive growth, with an audience that has doubled in size over the last year - from 2.4M developers in Q1 2021 to 5M in Q1 2022. This trend is largely attributed to Google’s decision to make Kotlin its preferred language for Android development. Kotlin is currently used by over a quarter of mobile developers, making it the third most popular language in mobile software projects behind Java and JavaScript.

The Rust community has nearly tripled in size in the last 24 months

Swift counts 3.5M developers and is currently the default language for development across all Apple platforms, which has led to a gradual phase-out of Objective C (2.4M) from the Apple app ecosystem. Nonetheless, Objective C has seen a slight uptick in its adoption for on-device IoT and AR/VR coding, which resulted in a small increase in its rank, from 12th to 11th place.

The more niche languages - Go, Ruby, Dart, and Lua - are still much smaller, with up to 3.3M active software developers each. Go and Ruby are important languages in backend development but Go has grown more than twice as fast in the past year in absolute terms, likely due to the fast development cycle it offers even though it is a compiled language. Dart has seen steady growth in the last year, propelled predominantly by the increasing adoption of the Flutter framework in mobile development. Finally, Lua was the third fastest growing language community in the last year, behind Rust and Kotlin, mainly attracting game and IoT developers looking for a scripting alternative to low-level languages such as C and C++.

Kotlin is another language we have consistently singled out for its impressive growth, with an audience that has doubled in size over the last year - from 2.4M developers in Q1 2021 to 5M in Q1 2022. This trend is largely attributed to Google’s decision to make Kotlin its preferred language for Android development. Kotlin is currently used by over a quarter of mobile developers, making it the third most popular language in mobile software projects behind Java and JavaScript.

The Rust community has nearly tripled in size in the last 24 months

Swift counts 3.5M developers and is currently the default language for development across all Apple platforms, which has led to a gradual phase-out of Objective C (2.4M) from the Apple app ecosystem. Nonetheless, Objective C has seen a slight uptick in its adoption for on-device IoT and AR/VR coding, which resulted in a small increase in its rank, from 12th to 11th place.

The more niche languages - Go, Ruby, Dart, and Lua - are still much smaller, with up to 3.3M active software developers each. Go and Ruby are important languages in backend development but Go has grown more than twice as fast in the past year in absolute terms, likely due to the fast development cycle it offers even though it is a compiled language. Dart has seen steady growth in the last year, propelled predominantly by the increasing adoption of the Flutter framework in mobile development. Finally, Lua was the third fastest growing language community in the last year, behind Rust and Kotlin, mainly attracting game and IoT developers looking for a scripting alternative to low-level languages such as C and C++.
Ranking of programming language communities over the last 24 months

1. JavaScript*
2. Python
3. Java
4. C/C++
5. C#
6. PHP
7. Kotlin
8. Swift
9. Go
10. Objective C
11. Rust
12. Ruby
13. Dart
14. Lua

* JavaScript includes CoffeeScript and TypeScript
Personality theories provide a blueprint for understanding why people behave the way they do. In the latest edition of our Developer Nation survey, we incorporated a measure of the widely accepted ‘Big Five’ personality dimensions in order to better understand the personality traits of software developers. Here, we share some of our findings and discuss how this kind of information can help to support interactions with developers.
Personality measures are a powerful tool for understanding people’s preferences and behaviours. Software teams need diversity not only in terms of skills, experience, and knowledge, but also require a variety of personalities to collaborate effectively on complex and challenging projects.

We used the Ten-Item Personality Inventory (TIPI) methodology in order to measure the ‘Big Five’ personality dimensions. These dimensions are: emotional stability, extraversion, openness to experiences, agreeableness, and conscientiousness. The TIPI method is well-suited for situations where short measures are required, and the results have been shown to have good alignment with other widely used Big Five measures. Although more comprehensive and accurate personality measures than TIPI exist, they typically require an entire survey to themselves.

The TIPI method presents respondents with ten pairs of personality traits and asks them to rate how strongly these traits apply to them. Below, we show responses to these items for over 12,000 developers. We find that developers, in general, see themselves as complex and open to new experiences (86% agree or strongly agree that this applies to them), dependable and self-disciplined (79%), calm and emotionally stable (76%), and sympathetic and warm (74%).

Developers are most likely to agree that they are dependable, self-disciplined, and open to new experiences

% of developers agreeing or disagreeing with each statement (Q1 2022 n=12,534)

<table>
<thead>
<tr>
<th>Trait</th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Neither agree nor disagree</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open to new experiences, complex</td>
<td>2%</td>
<td>10%</td>
<td>44%</td>
<td>42%</td>
<td>5%</td>
</tr>
<tr>
<td>Dependable, self-disciplined</td>
<td>5%</td>
<td>15%</td>
<td>49%</td>
<td>30%</td>
<td>6%</td>
</tr>
<tr>
<td>Sympathetic, warm</td>
<td>5%</td>
<td>19%</td>
<td>51%</td>
<td>23%</td>
<td>6%</td>
</tr>
<tr>
<td>Calm, emotionally stable</td>
<td>6%</td>
<td>18%</td>
<td>49%</td>
<td>27%</td>
<td>6%</td>
</tr>
<tr>
<td>Reserved, quiet</td>
<td>6%</td>
<td>14%</td>
<td>24%</td>
<td>39%</td>
<td>17%</td>
</tr>
<tr>
<td>Extraverted, enthusiastic</td>
<td>6%</td>
<td>16%</td>
<td>27%</td>
<td>35%</td>
<td>17%</td>
</tr>
<tr>
<td>Critical, quarrelsome</td>
<td>19%</td>
<td>28%</td>
<td>27%</td>
<td>21%</td>
<td>5%</td>
</tr>
<tr>
<td>Anxious, easily upset</td>
<td>21%</td>
<td>34%</td>
<td>23%</td>
<td>17%</td>
<td>5%</td>
</tr>
<tr>
<td>Conventional, uncreative</td>
<td>26%</td>
<td>38%</td>
<td>22%</td>
<td>11%</td>
<td>3%</td>
</tr>
<tr>
<td>Disorganised, careless</td>
<td>32%</td>
<td>36%</td>
<td>19%</td>
<td>11%</td>
<td>3%</td>
</tr>
</tbody>
</table>
Diving deeper into the TIPI data allows us to identify more specific personality types within the general developer population. We collapsed these ten items into five distinct measures, one for each of the Big Five personality dimensions. For example, statements about being ‘sympathetic, warm’ and ‘critical, quarrelsome’ combine to give an overall measure of agreeableness. We then derived a score for each developer on each of the five dimensions and identified the developers at the polar ends of each dimension, e.g. labelling those who are at the top end of the agreeableness scale as ‘agreeable’ and those at the bottom end as ‘disagreeable’. Finally, we segmented all developers into a set of distinct personality types, using the personality labels that they had been assigned as inputs to our segmentation algorithms.

The following charts show the characteristics of five example personality profiles revealed within our data. A well-rounded, ‘balanced’ personality type accounts for 52% of the developer population. These are developers who sit firmly at the centre of each dimension; neither introverted nor extroverted, highly agreeable nor disagreeable, emotionally unstable nor lacking emotion, etc.

Approximately 8% of all developers differ from the aforementioned group, with a higher level of openness to experiences – often related to intellectual curiosity. These software developers have personality traits that suggest they are likely to investigate new tools and technologies and stay up to date with the cutting edge of technology.

5% of developers fit a ‘responsible and cooperative’ personality type. These developers score highly in conscientiousness, openness to experiences, and agreeableness in comparison to the majority of developers. Increased conscientiousness often relates to setting long-term goals and planning routes to achieve them – e.g. being career-driven – while higher scores for openness to experiences reflects a preference for creativity and flexibility rather than repetition and routine. Our data backs this up – these developers are more receptive to personal development-related vendor resources, e.g. 35% engage with seminars, training courses, and workshops compared to 25% of ‘balanced’ developers. Their high scores for agreeableness also correlate with greater engagement with community offerings, e.g. 23% attend meetup events compared with 17% of ‘balanced’ developers.
5% of developers conform to an ‘achievement-driven and emotionally stable’ profile. As with the previous personality type, they are conscientious and open to experiences, however, they score much higher in terms of emotional stability but slightly lower in terms of agreeableness. Developers who score high in emotional stability react less emotionally – e.g. favouring data over opinions – while lower agreeableness can be a useful trait for making objective decisions, free from the obligation of pleasing others.

We also find a segment of developers with an ‘introverted and unreliable’ profile. These are independent personality dimensions (extraversion/introversion is not correlated with conscientiousness) -- there are certainly many developers who have high conscientiousness scores alongside low extraversion scores. However, the segment we show here scores particularly low in terms of ‘conscientiousness’ - which is an indicator that they may be less reliable. In addition they score low in extraversion. Developers in this profile are less involved in social activities, disorganised, closed to new experiences, and less agreeable than other developers. Fortunately, these developers, who are likely hard to reach and engage in new activities and communities, are a very small minority, at 2% of all developers.
Common developer personality profiles
(Q1 2022 n=12,534)

- **Responsible & cooperative** (5% of all developers)
- **Achievement-driven & stable** (5% of all developers)
- **Introverted & unreliable** (2% of all developers)

*Personality dimensions are independent. Each personality dimension does not necessarily correlate with another (e.g. extraversion is not correlated with conscientiousness).*
Finally, we show how the characteristics of these profiles vary, in terms of both associations with developer roles and the kinds of information and content that they consume. Developers in the ‘balanced’ profile are most likely to have ‘programmer/developer’ job titles. However, those who fit the ‘responsible and cooperative’ profile are disproportionately more likely to occupy creative (e.g. UX designer) roles, aligning with their increased creativity/openness, and senior CIO/CTO/IT manager positions, reflecting their self-discipline and achievement striving.

Those who are ‘achievement-driven and emotionally stable’ are less likely than other personality types to have ‘programmer/developer’ job titles, but disproportionately more likely to be data scientists, machine learning (ML) developers, or data engineers – further reinforcing the notion that they deal mainly in facts and data rather than opinions and emotions. Those in the ‘introverted and unreliable’ profile are more likely to have test/QA engineer and system administrator job titles than those in other personality types.
Achievement-driven developers with high emotional stability are 50% more likely to be data scientists than those with a balanced personality

% of developers with each personality type that hold each role (Q1 2022 n=12,531)

<table>
<thead>
<tr>
<th>Role</th>
<th>Personality</th>
<th>Balanced</th>
<th>Intellectually curious</th>
<th>Responsible &amp; cooperative</th>
<th>Achievement-driven &amp; stable</th>
<th>Introverted &amp; unreliable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmer / software developer (incl. frontend, backend, full-stack)</td>
<td></td>
<td>51%</td>
<td>58%</td>
<td>48%</td>
<td>43%</td>
<td>51%</td>
</tr>
<tr>
<td>Architect (system/solution/software/app)</td>
<td></td>
<td>10%</td>
<td>17%</td>
<td>10%</td>
<td>8%</td>
<td>10%</td>
</tr>
<tr>
<td>Data scientist, ML developer, or data engineer</td>
<td></td>
<td>8%</td>
<td>12%</td>
<td>11%</td>
<td>12%</td>
<td>8%</td>
</tr>
<tr>
<td>CIO / CTO / IT manager</td>
<td></td>
<td>7%</td>
<td>11%</td>
<td>12%</td>
<td>10%</td>
<td>8%</td>
</tr>
<tr>
<td>DevOps engineer/specialist</td>
<td></td>
<td>6%</td>
<td>9%</td>
<td>4%</td>
<td>6%</td>
<td>5%</td>
</tr>
<tr>
<td>UX designer</td>
<td></td>
<td>4%</td>
<td>6%</td>
<td>8%</td>
<td>6%</td>
<td>3%</td>
</tr>
<tr>
<td>Test/QA developer or engineer</td>
<td></td>
<td>4%</td>
<td>5%</td>
<td>4%</td>
<td>4%</td>
<td>9%</td>
</tr>
</tbody>
</table>

<5pp below the average of other personalities
2.5 – 5pp below the average of other personalities
±2.5pp around the average of other personalities
2.5 – 5pp above the average of other personalities
>5pp above the average of other personalities
When it comes to where developers go to find information and stay up to date, perhaps unsurprisingly, the ‘introverted and unreliable’ personality type uses the fewest information sources overall, affirming that they are a difficult group to engage via community-focused events and groups. However, their use of social media is in line with other personality types, suggesting that this may be a suitable channel for catching the attention of this hard-to-reach group.

Both of the high-conscientiousness and high-openness personality types use the widest range of information sources overall, however, those who are more cooperative are considerably more likely to turn to social media for information about software development (53% of the ‘responsible and cooperative’ type vs. 44% of the ‘achievement-driven and emotionally stable’ type).

‘Intellectually curious’ developers are the most likely to make use of official vendor resources and open source communities. Hence, the audience that vendors reach via these resources may be slightly more keen to experience new products and offerings, than the typical ‘balanced’ developer.

In this chapter, we just began to scratch the surface of developers’ personality profiles. The personality types we have shown are indicative of just a few of the differences that exist among developers. By capturing this kind of data, we’ve opened the door for more extensive profiling and persona building, along with a deeper analysis of how the many other developer behaviours and preferences that we track align with personality traits. If you’re interested in learning more about developer personalities and how this can help you to reach out to developers, then we’re very excited to see how our data can support you.
Achievement-driven developers use more information sources than those with a balanced personality

% of developers with each personality type (Q1 2022 n=11,673)

<table>
<thead>
<tr>
<th>Information sources used</th>
<th>Personality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seminars, training, or workshops</td>
<td>Balanced</td>
</tr>
<tr>
<td>Open source communities (e.g. GitHub)</td>
<td>25%</td>
</tr>
<tr>
<td>Official vendor newsletters</td>
<td>19%</td>
</tr>
<tr>
<td>Community groups (e.g. Google groups, Slack, mailing lists)</td>
<td>34%</td>
</tr>
<tr>
<td>Official vendor websites (docs, blogs, forums)</td>
<td>42%</td>
</tr>
<tr>
<td>Meetups</td>
<td>17%</td>
</tr>
<tr>
<td>Social media (e.g. Twitter, Facebook)</td>
<td>46%</td>
</tr>
</tbody>
</table>

- <Spp below the average of other personalities
- 2.5 – 5pp below the average of other personalities
- ≈2.5pp around the average of other personalities
- 2.5 – 5pp above the average of other personalities
- >5pp above the average of other personalities
Low-code/no-code (LCNC) tools provide a visual approach to software development, abstracting and automating parts of the application development process. This allows those without prior software development experience to create custom applications and provides potential time- and cost-saving for professional developers. In this chapter, we investigate the extent to which developers are using LCNC tools, showing differences according to professional status, geographical regions, and experience levels.
When it comes to reducing development overheads, addressing the challenge of finding skilled developers, and accelerating taking software to market, LCNC tools are becoming increasingly attractive. The sophistication of these tools is increasing rapidly, providing the potential to significantly disrupt the software industry. This begs the question, to what extent are developers’ using LCNC tools for their development projects?

We begin by separating developers according to their professional status – differentiating professionals from non-professionals, who are hobbyists and/or students. We excluded from our sample those who indicated that they were unsure about what share of their development work was done using LCNC tools. Just over half (54-55%) of developers in each group report that they are not using LCNC tools at all for their development work. This proportion is marginally lower for non-professionals who are students (55% of those who are exclusively students and 53% who are students and hobbyists) than non-professionals who identify as exclusively hobbyists (57%).

The proportion of developers who do use LCNC tools does not differ across groups (46% of professionals vs 45% of non-professionals). This highlights that LCNC tools are finding traction among those less likely to be familiar with coding and that use-cases within professional software development are also common.

1 We characterise all survey respondents who self-identify as being involved in software development as “developers”. Many developers have overlapping technical and non-technical/business roles. Although we comprehensively cover those with technical roles, we do not claim to fully represent non-technical/business audiences, such as those included under the broad term “citizen developers”.

OF PROFESSIONAL DEVELOPERS USE LOW-CODE/NO-CODE TOOLS FOR SOME PORTION OF THEIR DEVELOPMENT WORK

46%
Although almost half of all developers engage with LCNC tools, the majority of this group are likely to only use them for a small proportion - less than a quarter - of their software development work, regardless of their professional status.

Only 11% of all developers use LCNC tools for more than half of their development work, demonstrating that these tools have a considerable way to go before replacing developers’ reliance on manual coding.
As experience increases, developers are less likely to use LCNC tools at all. This is particularly true among those with more than ten years of experience. These tools are often framed as being best suited for simple programming tasks. Hence, the complexity of development work assigned to more experienced developers may be less appropriate for LCNC approaches. Furthermore, experienced developers are likely to have mastery over simpler coding tasks, which leaves little room for the efficiency gains that LCNC tools are often heralded for.

Using LCNC tools without a degree of accompanying manual coding is highly uncommon across all experience levels. The proportion of developers who use LCNC tools for a small amount (up to a quarter) of their development work remains relatively constant (between 17-24%) across the experience spectrum. Therefore, LCNC’s most likely role is as an occasional adjunct to existing coding tools, regardless of developers’ experience.

Experienced developers, particularly those with more than 10 years of experience, are the least likely to use LCNC tools

More extensive use of LCNC tools, i.e. for between one-quarter and three-quarters of all development activity, peaks slightly for those with around three to ten years of experience, revealing that it is early to mid-experience developers, rather than newcomers who are most likely to elevate LCNC tools’ status to essential. This is perhaps due to the recognised career importance of gaining traditional development experience, before reducing reliance on writing code. Only 2-4% of developers across all experience levels use LCNC tools for 75% or more of their development tasks, indicating that it is highly uncommon to shift the balance heavily towards LCNC-driven development.
Developers’ reliance on LCNC tools drops sharply with experience

% of developers in each experience level (Q1 2022 n=16,043)

- I do not use low-code or no-code tools
- Less than 25% of development work
- 25% to 49% of development work
- 50% to 74% of development work
- 75% or more of development work
Our data reveal notable differences in adoption and engagement with LCNC tools across different geographic regions. The Greater China area emerges as the region in which developers are most likely to be using LCNC approaches. 69% of developers in this region report using LCNC tools, compared to the global average of 46%. This suggests that the Chinese LCNC tool market has transitioned from an introduction phase to a growth phase. According to Mendix’s State of Low-Code report, IT professionals in China are the most likely to suggest that low-code is a trend their organisation can’t afford to miss (84% compared to 72% globally). Non-developer, or citizen developer, audiences also likely account for a large part of LCNC’s growth. However, as in all regions, the majority of bona fide software developers in the Greater China area currently use LCNC tools for less than half of their overall development work. It remains to be seen whether their reliance on such tools will also expand as the market and tools mature.

North America has the second highest LCNC tool adoption rate and stands out for the proportion of developers using LCNC tools to conduct more than half of their overall development work – 19% of developers here report that their use of LCNC tools outweighs their manual coding (comprising 13% using them for half to three-quarters of development work and 6% using them for more than three-quarters); almost double the global average of 10%. Hence, North America appears to be at the forefront of the LCNC movement, providing the strongest evidence that these tools can supplant traditional development approaches - even in a region where 81% of developers identify as professionals.
South Asia, the Middle East and Africa, and East Asia excluding Greater China are all above the global average in terms of LCNC tool adoption. Despite considerable uptake in these regions, LCNC products have not matured to the point where their use is a dominating feature of developers’ processes. Regions such as Western Europe and Israel, Oceania, Eastern Europe, and South America are all below the global average in terms of LCNC tool adoption.

The shortfall in these regions is particularly linked to smaller than average proportions using LCNC tools for more than 25% of their development work. The proportion using them for less than a quarter of their work is more comparable to the global average, suggesting that the market is still in its introductory phase in these regions – developers are evaluating the tools but are yet to rely on them for a substantial portion of their work.
In this chapter, we focus our attention on some of the key differences between developers in East Asia, including the Greater China region, and the rest of the world. Understanding these differences provides valuable insights that can help shape the strategy for developer engagement programs.
For this analysis, we split the Greater China area from the rest of East Asia to provide more regional granularity. In terms of relative size, we find that almost a fifth (18%) of the global developer population is located in either the Greater China region (9%) or the rest of East Asia (9%). Breaking down East Asia into countries, we see that more than half of the developers here are spread across two countries: Indonesia (32%) and Japan (21%).

When comparing developers across regions, we can see that just over a third (34%) of developers in the Greater China region have six or more years of experience, which is notably less than developers globally (43%). Furthermore, the Greater China region has a much smaller concentration (4% vs 22% globally) of highly-experienced developers (16+ years). With generally lower levels of experience in the Greater China area, aspiring developers may find starting a career here less competitive than developers in regions with higher levels of experience.

East Asian developers outside China have similar levels of experience to the rest of the global developer population. Both groups have a little more than a third (34%) of their developers with 11+ years of software development experience. However, East Asia's data are largely propped up by Japan. The developer community in Japan tends to be highly experienced, with almost six in ten developers (59%) having 16+ years of experience. No other country has a higher concentration of developers with this level of experience. With such a high concentration of highly skilled developers, we can expect some differences in behaviour, which we'll highlight in the last section of this chapter.
Experience in software development by region
% of developers in each region (Q1 2022 n=20,037)
The journey to coding mastery lacks a clearly defined path. Developers typically state they’ve used more than two learning methods on average to learn how to code. In general, the self-taught method is the most popular among developers globally, with more than 60% using this method. However, our data shows that the proportion of self-taught developers fluctuates significantly across regions.

In the Greater China area, the most popular method for developers to learn how to code is via an undergraduate degree in computing, with 50% having used this method. This is significantly higher than developers in other regions (41% - 42%). We generally see a higher concentration of professional developers in Greater China (83%) than we do in the rest of the world (70%). It could be that the job market in Greater China more often requires a degree in computing or engineering, which would also explain why self-teaching is used less often in this region.

Developers in the rest of East Asia, however, tend to follow the learning trends of developers in other regions. Here, we see the self-taught method is the most popular method (61%), followed by an undergraduate degree in software engineering (41%). Analysing the data at a country level, we see developers in Indonesia are more diverse learners. Developers in this country stated that they used three methods on average when learning to code. Indonesian developers are more likely to learn via self-teaching, online courses, and developer bootcamps than any other developers in East Asia. This is quite different from their peers in Japan who are the least likely to use online courses and bootcamps to learn how to code. Instead, developers in Japan are most likely to use the self-taught (63%) and on-the-job training (45%) methods when learning to code.
### Where developers learned to code

% of developers having used each method for learning to code (Q1 2022 n=13,902)

<table>
<thead>
<tr>
<th>Method</th>
<th>East Asia developers</th>
<th>Greater China developers</th>
<th>Other developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-taught</td>
<td>61%</td>
<td>41%</td>
<td>64%</td>
</tr>
<tr>
<td>Undergraduate degree in computing / software engineering</td>
<td>41%</td>
<td>50%</td>
<td>39%</td>
</tr>
<tr>
<td>Online courses (e.g. Coursera, Lynda.com, Udemy)</td>
<td>34%</td>
<td>18%</td>
<td>38%</td>
</tr>
<tr>
<td>On-the-job training</td>
<td>27%</td>
<td>19%</td>
<td>31%</td>
</tr>
<tr>
<td>Developer boot camp</td>
<td>21%</td>
<td>12%</td>
<td>15%</td>
</tr>
<tr>
<td>Other classroom training (non-degree)</td>
<td>14%</td>
<td>9%</td>
<td>12%</td>
</tr>
<tr>
<td>Other degree including programming courses</td>
<td>14%</td>
<td>8%</td>
<td>14%</td>
</tr>
<tr>
<td>Postgraduate degree in computing / software engineering</td>
<td>9%</td>
<td>13%</td>
<td>19%</td>
</tr>
</tbody>
</table>
Our focus on developers in East Asia and the Greater China area shows Stack Overflow’s popularity falls below the global average. Developers in these regions are around three times less likely to visit Stack Overflow than developers in other regions. Developers in the Greater China area are the least engaged, with only 19% having an account, and only 11% having earned at least one badge. Developers in this region have other home-grown Q&A site alternatives, such as segmentfault.com, which could be contributing to the lower adoption of Stack Overflow.

When looking closely at the rest of East Asia, we again see that developers in Japan are skewing the perception of this region. Developers in Japan have even less activity on Stack Overflow than developers in the Greater China area. Here, only a little more than a third (36%) stated they use Stack Overflow. Furthermore, only about 5% have an account. Like developers in the Greater China area, Our data does show usage of Stack Overflow increases among Japanese developers who have gained experience in software development, indicating that less experienced developers are using other platforms for support. Like China, Japan has other home-grown options like teratail.com, where developers can field programming support from their peers, which may be the place new Japanese programmers visit more often to get answers to their questions.

Next, we explore how developers interact with the popular online community, Stack Overflow, to understand their engagement levels with programming support. Stack Overflow has become a standard support community for many developers, with more than eight in ten (85%) of the general developer population reporting they’ve used or visited this popular question and answer site.

Tip: You can find more regional data and apply filters in the SlashData Dashboards: https://sdata.me/SDashboards
Developers’ activity on Stack Overflow
% of developers in each region Q1 2022 n=10,302

- **I don’t use/visit Stack Overflow**
  - South Asia: 25%
  - Greater China: 13%
  - Rest of world: 10%

- **I visit Stack Overflow, but I don’t have an account**
  - South Asia: 40%
  - Greater China: 43%
  - Rest of world: 34%

- **I have an account on Stack Overflow, but no badges**
  - South Asia: 25%
  - Greater China: 19%
  - Rest of world: 29%

- **I have an account and I have earned at least one badge**
  - South Asia: 11%
  - Greater China: 10%
  - Rest of world: 24%
How businesses and developers as individuals make money from software projects is one of the most important decisions they have to make. Of all the business models and strategies available, companies and freelancers need to pick the ones that best match their market and goals. This chapter focuses on the popularity of revenue models among professional developers and the companies they work for.
Of all the revenue models we track in our surveys, contracted development / consulting is the most popular model. As of Q1 2022, 31% of professional developers are using this model, 7 percentage points more than the next closest revenue model - selling apps or software. Contracted development can span months or even years, allowing for developers and companies to properly plan out resources during the project. In addition, professional developers and their companies may find the clients they contract for require additional services, thus leading to additional revenue. Contracted development is tried-and-true as it’s been the most popular revenue model for the past five surveys.

Selling apps/software through an app store or their own portal is the second most popular revenue model, with almost a quarter (24%) of professional developers making money in this way. Furthermore, adoption of this model has been stable over two and a half years, despite “Epic” lawsuits against Apple and Google in 2021, which argued that these app stores had excessive fees and restrictive payment collection processes. App stores and portals are popular now, but other technologies, such as progressive web apps (PWAs), could start to impact the popularity of app stores. PWAs can work across multiple platforms, provide a native experience, and can help developers avoid high commission fees from app stores; all of which are big incentives to embrace the power of the web.

Interestingly, less than a tenth (7%) of professional developers are generating revenue by selling data. Data has often been referred to as the new gold and data breaches are heavily covered in news articles as well. If data is so valuable, why are so few professional developers using this model? Regulatory measures, such as the EU’s General Data Protection Regulation (GDPR), could be hampering developers’ ability to sell user data based on the “right to be informed” principle. The California Consumer Privacy Act (CCPA) also has multiple restrictions for selling user data including an earnings cap based on a company’s total revenue. These are just a couple of examples of why selling data is difficult, which impacts its popularity as a revenue model.
How professional developers and their organisations generate revenue

% of professional developers using each revenue model (Q1 2022 n=10,299)

- Contracted development / research / consulting / design: 31%
- Selling apps / software / solutions through an app store or own portal: 24%
- Subscriptions: 21%
- Software licenses: 20%
- Selling services / tools or APIs to developers: 20%
- e-Commerce sales: 14%
- In-app purchases: 14%
- Advertising: 12%
- Selling physical non-tech products that work with your apps / software: 10%
- Selling hardware components or devices: 10%
- Indirectly (brand awareness, user/product recommendations, cost savings): 9%
- Projects facilitate/enable other business activities, but aren’t revenue-generating: 7%
- Selling data: 7%
- Per-device licencing or royalty fees: 7%
- Selling filmmaking or animation services: 4%
- Not sure / not yet decided: 5%
- We’re a non-profit or government entity, not seeking profit: 4%
- Not interested in revenue from my/our projects: 3%

Revenue-generating methods
Non-revenue-generating methods
Next, we will look at how the industries that developers are active in influence their revenue models. Contracted development is the most popular revenue model across all sectors, further emphasising the effectiveness of this model.

Developers active in the software products and services, data analytics, and financial services verticals tend to have the same revenue strategies. Professional developers in all three of these sectors have the same top-three revenue model choices. In addition to contracted development, app stores and selling services/APIs are the more popular methods for generating revenue in these sectors.

In-app purchases break into the top three among developers in the entertainment and media sector. 28% of professional developers in this vertical are using this method, double the percentage of the general developer population. In-app purchases are strongly associated with the freemium strategy where users are able to use/download applications for free with some features restricted to micro-purchases. This strategy has become quite popular in game development for building a base of users and incrementally generating revenue, as long as the quality of production is high.

For professional developers working for companies in the marketing and advertising sectors, the advertising revenue model rises to second place, but it’s unable to unseat contracted development as the most used model. Looking across industries, there’s an apparent lack of usage of advertising as a revenue model among most other developers. On average, advertising is ranked eighth among professional developers outside of the marketing and advertising industry, being used about three times less often. Again, privacy protection may be hindering developers’ ability to use this revenue model effectively.
### Ranking of revenue models by industry vertical

Top five revenue models in each industry vertical (Q1 2022 n=10,245)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Software products and services</th>
<th>Data analytics / Business intelligence</th>
<th>Financial services &amp; banking</th>
<th>Education, training, and research</th>
<th>Telecommunications and networks</th>
<th>Entertainment, media, and information</th>
<th>Health, medical, biotechnology, and pharmaceuticals</th>
<th>Marketing and advertising services</th>
<th>Manufacturing</th>
<th>Hardware products</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Contracted development</td>
<td>Contracted development</td>
<td>Contracted development</td>
<td>Contracted development</td>
<td>Contracted development</td>
<td>Contracted development</td>
<td>Contracted development</td>
<td>Contracted development</td>
<td>Contracted development</td>
<td>Contracted development</td>
</tr>
<tr>
<td>2</td>
<td>Selling apps through store or own portal</td>
<td>Selling apps through store or own portal</td>
<td>Selling apps through store or own portal</td>
<td>Selling apps through store or own portal</td>
<td>Selling apps through store or own portal</td>
<td>Selling apps through store or own portal</td>
<td>Advertising</td>
<td>Software licenses</td>
<td>Selling hardware components or devices</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Selling services / APIs to developers</td>
<td>Selling services / APIs to developers</td>
<td>Selling services / APIs to developers</td>
<td>Subscriptions</td>
<td>Software licenses</td>
<td>In-app purchases</td>
<td>Subscriptions</td>
<td>Selling apps through store or own portal</td>
<td>Selling apps through store or own portal</td>
<td>Software licenses</td>
</tr>
<tr>
<td>4</td>
<td>Software licenses</td>
<td>Subscriptions</td>
<td>Selling services / APIs to developers</td>
<td>Selling services / APIs to developers</td>
<td>Subscriptions</td>
<td>Software licenses</td>
<td>Subscriptions</td>
<td>Selling services / APIs to developers</td>
<td>Selling apps through store or own portal</td>
<td>Selling apps through store or own portal</td>
</tr>
<tr>
<td>5</td>
<td>Subscriptions</td>
<td>Subscriptions</td>
<td>Software licenses</td>
<td>Subscriptions</td>
<td>Advertising</td>
<td>Selling services / APIs to developers</td>
<td>e-Commerce sales</td>
<td>Subscriptions</td>
<td>Selling services / APIs to developers</td>
<td></td>
</tr>
</tbody>
</table>

Sector developers' companies are active in
Finally, we evaluate revenue model usage among developers in different-sized companies. Again, contracted development remains the most popular model across every size of company. This strategy is the status quo for developers, and, with such popularity, it’s presumed to be the expectation by customers seeking professional development.

Developers working for micro-businesses are the most likely to report that they generate revenue from contracted development, with over a third (36%) of developers who work in them using this model. Professional developers in micro-businesses are also using multiple revenue models slightly more often than other developers. This indicates that companies of this size are trying to maximise their earning potential while relying heavily on the industry standard of contracted development. That being said, contracts don’t sell themselves, and micro-businesses have only 2-20 employees, so developers in these companies will likely be a close part of sales conversations.

Usage of the advertising revenue model declines as companies grow in size

Developers at large enterprises have a slightly different profile, as they tend to use the contracted development model less often than developers in other company sizes. We also see less use of the multiple revenue model, indicating that companies of this size have a more focussed strategy for generating revenue.

Freelance professionals are also heavily reliant on contracted development (28%), but we also see increased usage of the advertising revenue model. Here, we can see developers trying to increase their earnings by applying advertising to their apps and the software they’re creating. However, the adoption of advertising declines in larger companies. Developers in large enterprises (5K+ employees) are almost half as likely to use advertising than developers in micro-businesses.
# Revenue model usage by company size

% of developers working for companies of each size using each revenue model (Q1 2022 n=10,302)

<table>
<thead>
<tr>
<th>Revenue model</th>
<th>Freelancer</th>
<th>Micro business (2-20 employees)</th>
<th>Small business (21-250 employees)</th>
<th>Medium-sized business (251 - 5K employees)</th>
<th>Large enterprise (5K+ employees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contracted development / research / consulting / design</td>
<td>28%</td>
<td>36%</td>
<td>30%</td>
<td>28%</td>
<td>26%</td>
</tr>
<tr>
<td>Selling apps / software / solutions through an app store or own portal</td>
<td>21%</td>
<td>26%</td>
<td>23%</td>
<td>20%</td>
<td>17%</td>
</tr>
<tr>
<td>Subscriptions</td>
<td>15%</td>
<td>24%</td>
<td>21%</td>
<td>16%</td>
<td>15%</td>
</tr>
<tr>
<td>Selling services / tools or APIs to developers</td>
<td>14%</td>
<td>19%</td>
<td>19%</td>
<td>20%</td>
<td>17%</td>
</tr>
<tr>
<td>Software licenses</td>
<td>10%</td>
<td>17%</td>
<td>20%</td>
<td>17%</td>
<td>14%</td>
</tr>
<tr>
<td>e-Commerce sales</td>
<td>12%</td>
<td>15%</td>
<td>14%</td>
<td>12%</td>
<td>11%</td>
</tr>
<tr>
<td>In-app purchases</td>
<td>12%</td>
<td>13%</td>
<td>13%</td>
<td>13%</td>
<td>8%</td>
</tr>
<tr>
<td>Advertising</td>
<td>16%</td>
<td>13%</td>
<td>11%</td>
<td>8%</td>
<td>7%</td>
</tr>
<tr>
<td>Selling physical non-tech products that work with your apps / software</td>
<td>4%</td>
<td>8%</td>
<td>12%</td>
<td>11%</td>
<td>8%</td>
</tr>
<tr>
<td>Selling hardware components or devices</td>
<td>5%</td>
<td>8%</td>
<td>11%</td>
<td>10%</td>
<td>10%</td>
</tr>
</tbody>
</table>
Many high-profile emerging technologies are slated to change the world - self-driving cars and the Metaverse spring to mind. However, there are many more outside of the spotlight, such as AI-assisted programming, that will also have profound and far-reaching implications. Here, we explore how developers are involved with a range of emerging technologies.
As part of our commitment to understanding the landscape of software development, we have tracked developers’ engagement with and adoption of different emerging technologies over eight surveys across the last four years.

To understand developers’ relationships with these technologies, we asked them if they are working on, learning about, interested in, or not interested in a wide range of experimental or novel technologies, adding them as they appear and removing them from the list as they mature, as we did with DevOps in the Q1 2021 edition of this report.

We consider a developer who is interested in, learning about, or working on a technology as being ‘engaged’ with it. The adoption rate of a technology is the proportion of engaged developers who are currently working on it. In order to better understand engagement and adoption with these technologies on a macro level, we created four quadrants, defined by where the engagement and adoption rates of a technology sit relative to their medians:

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.

In this edition of our Developer Nation survey, we added several new technologies to our tracker:

- AI-assisted software development
- The Metaverse
- Non-fungible tokens (NFTs)
It’s immediately apparent that AI-assisted software development captures developers’ interest - the possible impacts on working practices, careers, and remuneration are especially salient to 67% of developers. This interest is not purely hypothetical or academic - 14% of engaged developers are actively working on AI-assisted software development, and adoption of this technology is the third-highest of any emerging technology. We can’t say for sure if developers are building or simply using these technologies, though, given their complexity and novel status, it’s likely that many of these adopters are using AI-assisted development as part of their workflow rather than actively developing the technology itself.

Developers who rely heavily on low- and no-code tools are the most likely to be working with AI-assisted software development

We are already seeing the effects of low- and no-code tools on the democratisation of software development. with 46% of developers reporting that they use such tools, they pervade beyond the citizen developer well into the professional realm. AI-assisted development is a logical addition for many developers looking to increase their development velocity, and indeed, we see that developers who do 75% or more of their development work using low- or no-code tools (20%) are four times as likely as those who don’t use them at all (4%) to be currently working on AI-assisted software development.
Further down the list, stalwarts such as computer vision, robotics, and blockchain (cryptocurrencies and other applications) command high levels of engagement amongst developers, though NFTs - another crypto-adjacent technology - has much lower engagement, with just 48% of developers working on, interested in, or learning about it. This said, the money-making potential of NFTs has not gone unnoticed by developers - 11% of those engaged report that they are currently working on the technology, making this a potentially profitable niche for those who do get involved. In fact, all three crypto-adjacent technologies have high adoption and learning rates - for each, at least 30% of engaged developers are actively learning about the technologies.

Blockchain technologies, including cryptocurrencies, have experienced the largest increase in engagement in the last 12 months, with interest in crypto currencies increasing by 14% and interest in non-crypto blockchain applications increasing by 15%, but adoption of this technology has stagnated, increasing by a single percentage point in the last 12 months.

Interestingly, we see that, compared to the previous year, growth in adoption rates has stagnated across the board. Part of this is due to the changing landscape of emerging technologies that we track, but careful examination of the change in engagement rates shows that many more developers are becoming engaged with a wider range of emerging technologies. In fact, the absolute adoption rates (the proportion of all developers working on a technology) have remained largely unchanged in the past year - developers have widened their interests but this has not yet trickled down to their working practices.
The Metaverse is another technology that has recently garnered a lot of interest, bounding into the public eye in October - likely coinciding with Facebook’s name change to Meta. We see that a healthy 53% of developers are engaged with this technology, but adoption is low, at 9% of engaged developers. This is likely because the Metaverse is still being defined.

The path to becoming a ‘Metaverse developer’ is unclear, and, being a confluence of several contributing hardware and software technologies - extended reality (XR), networking, graphics, optics, machine learning, and blockchain, to name a few - many of which have yet to reach maturity, lots of developers will be waiting to see what the future holds. Indeed, 28% of engaged developers say that they are currently learning about the Metaverse, one of the highest learning rates outside the crypto/blockchain space. Many of these developers are likely positioning themselves to make the most of a possibly lucrative new technology.
AI-assisted development is especially salient to developers

% of developers currently working on, interested in, or learning about each technology (Q1 2022 n=13,939)

- AI-assisted software development: 67%
- Computer vision: 62%
- Robotics: 61%
- Cryptocurrencies (e.g. Bitcoin): 59%
- Blockchain applications other than cryptocurrency: 59%
- Self-driving cars: 57%
- Quantum computing: 55%
- Drones: 54%
- The Metaverse: 53%
- Brain / body computer interfaces (e.g. advanced prosthetics, neural lace): 52%
- 5G / 6G: 51%
- Conversational platforms / voice search: 50%
- Non-fungible tokens (NFTs): 48%
- DNA computing / storage: 44%
- Fog / Edge computing: 41%
- Haptic feedback: 40%
- Hearables: 37%
Blockchain applications, cryptocurrencies, and NFTs have the highest share of developers learning about them

% of engaged developers working, learning about, or interested in each technology (Q1 2022 n=13,939)

<table>
<thead>
<tr>
<th>Technology</th>
<th>Adopted</th>
<th>Learning about</th>
<th>Interested in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptocurrencies (e.g. Bitcoin)</td>
<td>16%</td>
<td>34%</td>
<td>50%</td>
</tr>
<tr>
<td>Computer vision</td>
<td>15%</td>
<td>25%</td>
<td>60%</td>
</tr>
<tr>
<td>AI-assisted software development</td>
<td>14%</td>
<td>28%</td>
<td>59%</td>
</tr>
<tr>
<td>Conversational platforms / voice search</td>
<td>13%</td>
<td>25%</td>
<td>62%</td>
</tr>
<tr>
<td>Blockchain applications other than cryptocurrency</td>
<td>12%</td>
<td>30%</td>
<td>58%</td>
</tr>
<tr>
<td>5G / 6G</td>
<td>11%</td>
<td>23%</td>
<td>66%</td>
</tr>
<tr>
<td>Non-fungible tokens (NFTs)</td>
<td>11%</td>
<td>32%</td>
<td>58%</td>
</tr>
<tr>
<td>Fog / Edge computing</td>
<td>10%</td>
<td>25%</td>
<td>64%</td>
</tr>
<tr>
<td>Robotics</td>
<td>9%</td>
<td>24%</td>
<td>67%</td>
</tr>
<tr>
<td>DNA computing / storage</td>
<td>9%</td>
<td>21%</td>
<td>70%</td>
</tr>
<tr>
<td>Drones</td>
<td>9%</td>
<td>21%</td>
<td>70%</td>
</tr>
<tr>
<td>The Metaverse</td>
<td>9%</td>
<td>28%</td>
<td>63%</td>
</tr>
<tr>
<td>Haptic feedback</td>
<td>8%</td>
<td>22%</td>
<td>70%</td>
</tr>
<tr>
<td>Hearables</td>
<td>8%</td>
<td>22%</td>
<td>70%</td>
</tr>
<tr>
<td>Brain / body computer interfaces (e.g. advanced prosthetics, neural lace)</td>
<td>7%</td>
<td>21%</td>
<td>72%</td>
</tr>
<tr>
<td>Self-driving cars</td>
<td>7%</td>
<td>20%</td>
<td>73%</td>
</tr>
<tr>
<td>Quantum computing</td>
<td>6%</td>
<td>22%</td>
<td>71%</td>
</tr>
</tbody>
</table>
Interest in blockchain applications and cryptocurrencies has increased the most of any technology but adoption of blockchain applications has stagnated
Interest and adoption change from one year ago (Q1 2021 n=12,291 | Q1 2022 n=13,939)

<table>
<thead>
<tr>
<th>LOW ADOPTION</th>
<th>HIGH ADOPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interest</td>
<td>Adoption</td>
</tr>
<tr>
<td>LOW INTEREST</td>
<td>HIGH INTEREST</td>
</tr>
<tr>
<td>Quantum computing</td>
<td>12%</td>
</tr>
<tr>
<td>Drones</td>
<td>10%</td>
</tr>
<tr>
<td>Self-driving cars</td>
<td>11%</td>
</tr>
<tr>
<td>The Metaverse</td>
<td>N/A</td>
</tr>
<tr>
<td>Haptic feedback</td>
<td>12%</td>
</tr>
<tr>
<td>Brain/body computer interfaces (e.g. advanced prosthetics, neural lace)</td>
<td>12%</td>
</tr>
<tr>
<td>Hearables</td>
<td>12%</td>
</tr>
<tr>
<td>DNA computing/storage</td>
<td>10%</td>
</tr>
</tbody>
</table>

- ENGAGEMENT CHANGE BETWEEN Q1 2021 AND Q1 2022
- ADOPTION CHANGE BETWEEN Q1 2021 AND Q1 2022
The 22nd edition of the Developer Nation survey reached 20,000+ respondents from 166 countries around the world. As such, the Developer Nation series of surveys continues to be the most global independent research on mobile, desktop, industrial IoT, consumer electronics, embedded, third-party app ecosystems, cloud, web, game, augmented and virtual reality, and machine learning developers and data scientists combined, ever conducted. The report is based on a large-scale, online developer survey designed, produced, and carried out by SlashData over a period of ten weeks between December 2021 and February 2022.

Respondents to the online survey came from 166 countries, including major app and machine learning development hotspots such as the US, China, India, Israel, the UK, and Russia, even stretching all the way to Kenya, Brazil, and Jordan. The geographic reach of this survey is truly reflective of the global scale of the developer economy. The online survey was translated into eight languages in addition to English, namely simplified Chinese, traditional Chinese, Spanish, Portuguese, Vietnamese, Russian, Japanese, and Korean, and was promoted by more than 80 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.

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

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

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

For more information on our methodology please visit https://www.slashdata.co/methodology.
Are you curious about the future developer trends?
Let’s partner up!

- **Reach new developer audiences**
  Get your offering in front of our Developer Nation community with 100,000+ developers in 165 countries. Promote your products, tools, events, and developer programs via newsletter inclusions, goody bag sponsorship, content amplification on our blog, or brand placement in our free reports.

- **Get insights into global developer trends**
  Understand how your audience stacks against the global average by inviting your developer community to participate in our research. Get 25+ graphs with the key trends based on the data from our surveys, which will help you shape your developer outreach strategies.

LET’S TALK!
Contact us
partnerships@slashdata.co