



The State of Web Development

Findings and insights to revolutionize your
web strategy in 2024 and beyond



The Netlify State of Web Development 2023 is your guide to cutting-edge trends and forward-thinking strategies that are shaping the world of web development.

As we share our fourth annual report (previously the Jamstack Community Survey), it's clearer than ever: we are witnessing a web renaissance unlike anything we have previously seen. A new evolution of the web – where code meets creativity. Where the future is being written by the minute. This evolution will not only impact web development teams, but also marketers, product managers, IT professionals, c-suite, and entire organizations in defining how they prioritize and execute long-term strategies.

This report unpacks more than why Next.js is slipping in both usage and overall sentiment among survey respondents and which framework is poised to become the most popular (spoiler alert: It's Astro). It also includes how the future of web development is being driven by emerging technologies and concepts that will help enterprises best position themselves for success in the future by making the most informed decisions today.

This report peels back the layers of modern web development, offering a glimpse of an industry undergoing significant transformation. Whether you're battling the limitations of monolithic architectures or just curious about trending web frameworks, this report is your roadmap to understanding and leveraging the latest developments in our field.

In this report, you'll find:

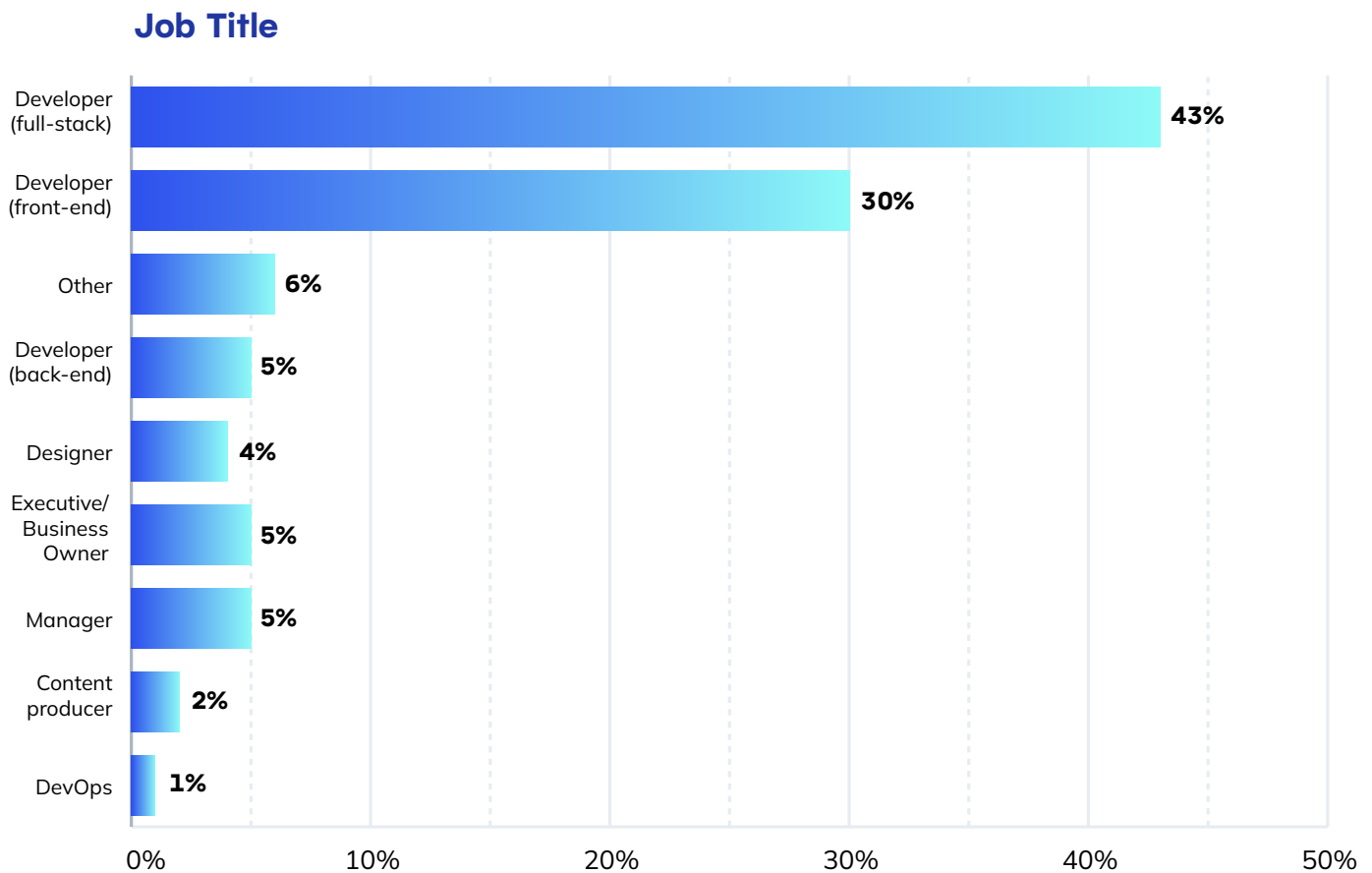
- Why composable architecture is more than a framework—it's a business state of mind
- Why generative AI will make it harder for monoliths to keep up
- Why Astro is gaining traction in both usage and satisfaction
- Why composable is the architecture of choice for B2B marketing, ecommerce, consumer software, and education sites
- How Netlify can help

Who's doing the building?

Almost 7,000 people participated in our survey this year and leapt at the chance to weigh in on the state of web development.

Job titles

Similar to previous years, a vast majority (78%) that participated claimed the job title of Developer, with “full stack developer” the most prominent. Executives and Managers comprised 9.6% of participants. Of the survey respondents, more than half (57%) are either self-employed or report full-time employment in these roles, and (74%) have been working in web development for longer than 2 years.



Why composable architecture is more than a framework, it's a business state of mind

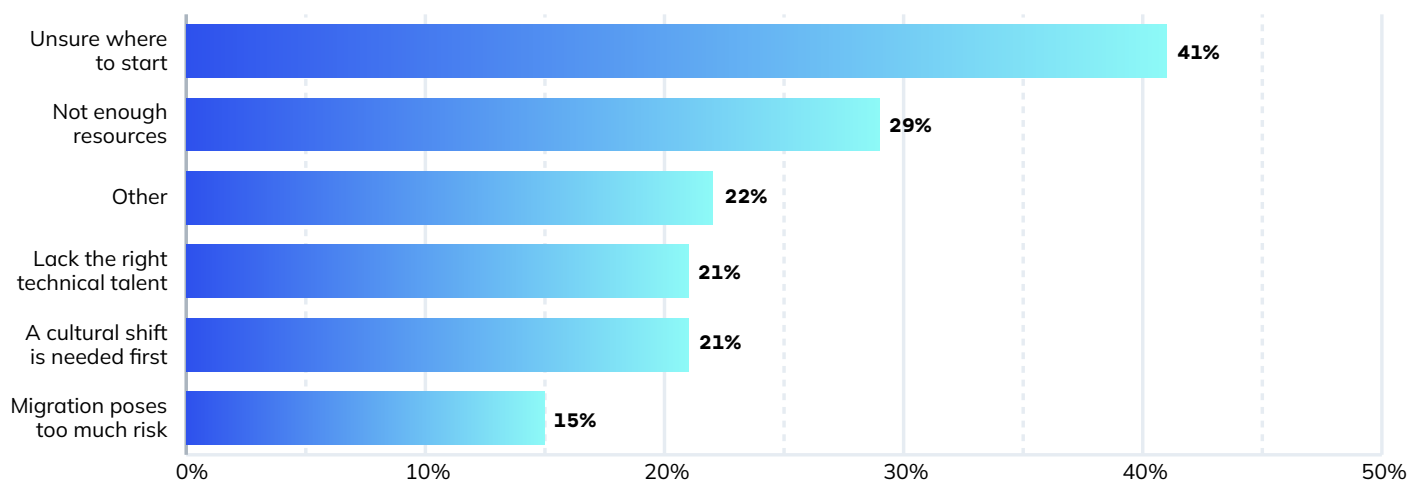
The advent of composable architectures is the natural destination for an industry hyper-focused on freedom, flexibility, interoperability, cohesiveness, performance, and productivity.

Composable architectures began with the deconstruction of all-in-one, or monolithic systems. Companies first went headless, which is often seen as the first step in adopting modular and customizable—composable—architectures. Removing the frontend, or presentation layer, from the monolith gave developers freedom to select their preferred frameworks and content management systems, attain unparalleled productivity, and gain immediate access to the performance and scalability of serverless technologies and architectures.

Now, composable has become so much more than a headless project. The data confirmed that 75% of all survey respondents had utilized composable architectures on the sites they've built in the last 12 months.

While many respondents leaned on composable architectures for their websites, the barriers to adopting it varied. 41% said they would build with composable right now if they had a clear path of where to begin. 29% reported that key resources were completely absorbed with maintaining their current, monolithic architecture and lacked the resources to make the switch. The most logical conclusion we can draw is that a majority of developers are building with composable architectures, but most likely are limited to personal or small, proof-of-concept projects as monoliths still rule the enterprise. However, many organizations want to start embracing composable architectures, but don't know where to start. The more developers build with composable tools, the more expertise will be created within organizations. But until organizations have the confidence to fully commit to composable, resources will be restricted to maintaining existing systems.

What barriers have prevented you switching to composable sites?



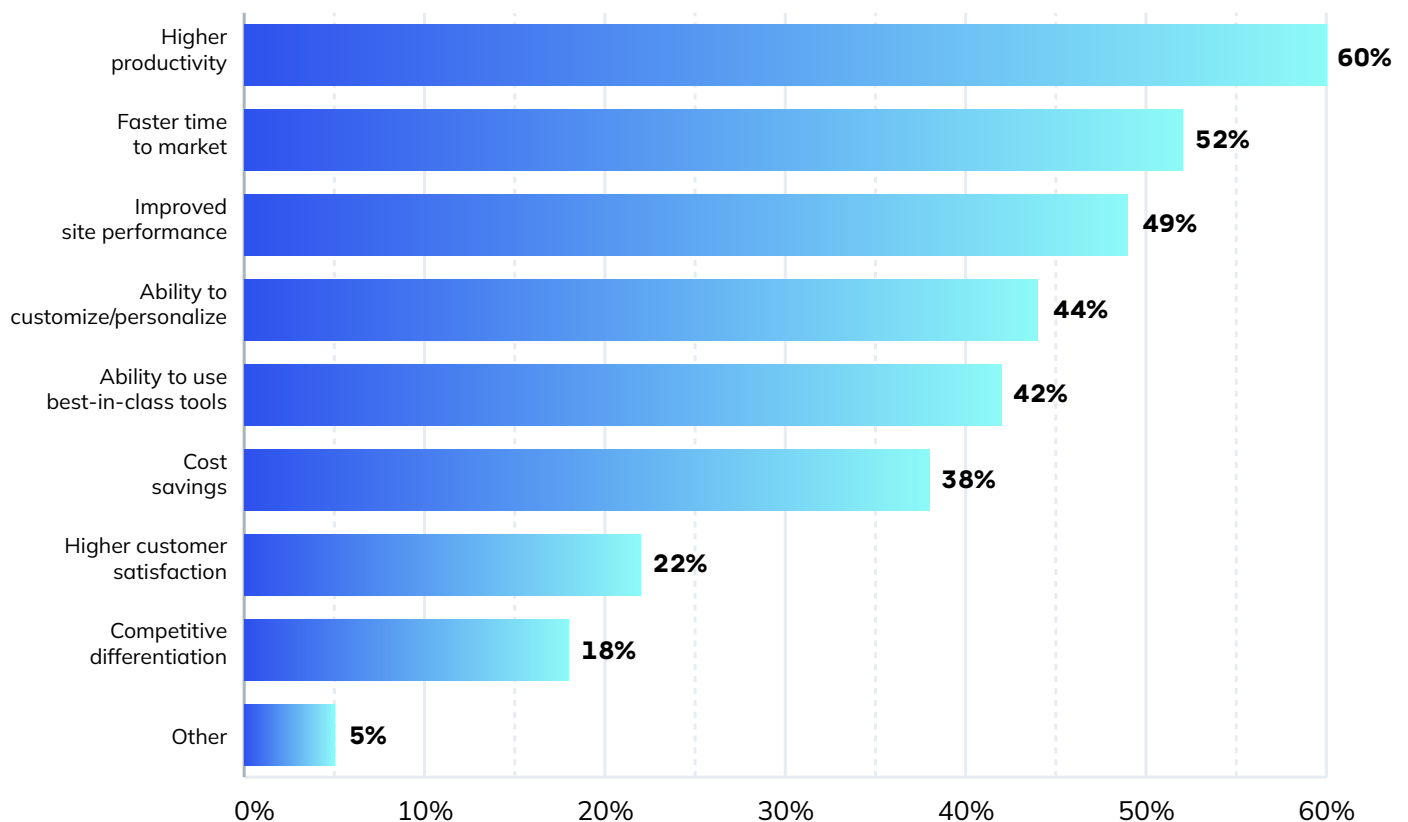
Interestingly, those who are actively building composable sites reported additional bandwidth and resource availability. Here's what else respondents had to say:

- **60%** report high productivity
- **52%** are realizing a faster time-to-market
- **49%** have seen a noticeable increase in site performance
- **44%** have a greater ability to customize and personalize their customer experiences
- **38%** have already experienced cost savings in less than 12 months of building these projects

When building websites for the web using monolithic architectures, there used to exist a “pick and choose” process during product evaluation. For the uninitiated who haven't sat through these types of process meetings before, they go something like this: teams will define technical features that are non-negotiable needs from their vendor of choice, then a list of features that are nice to have, but not required, and followed by a long list of features that the organization will probably never use.

Composable architectures are top of mind for any enterprise use cases

What benefits have you seen when building composable sites vs. legacy architectures?



With a monolith, freedom was always limited as software solutions were selected with the clear understanding that they were never meant to be an ideal match to your needs. All of that changes with composable architectures.

Now, organizations can pick and choose every aspect of their web stacks and every single component is meticulously selected as a best-in-class solution for that part of the stack. Need the top-rated CMS integrated with a friendly payment provider built using Astro and served on Netlify's CDN? No problem.

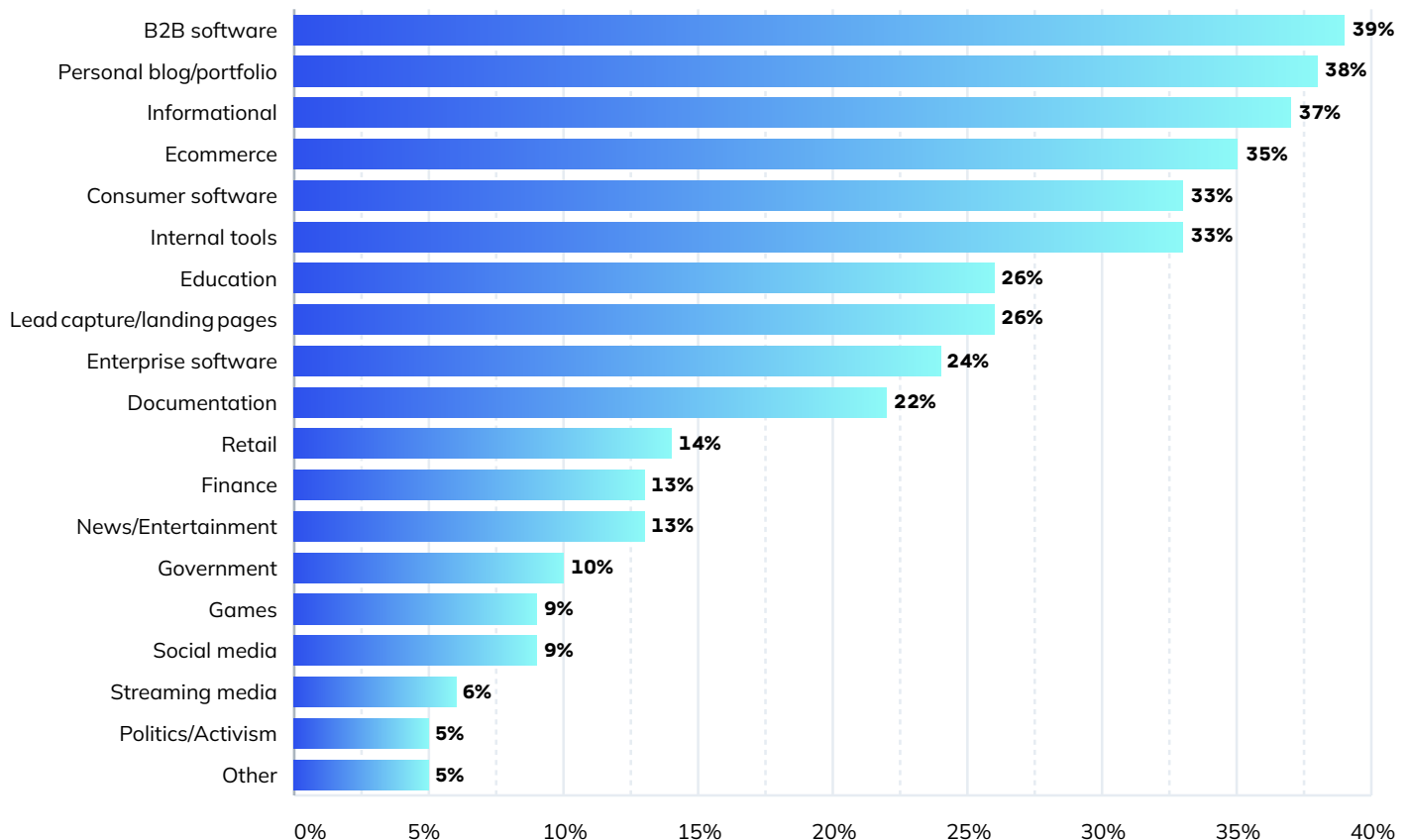
The developer experience has long been at the heart of Netlify's vision. Accelerating developers and providing the best experience not only means

developers accomplish more, but they also have the freedom to customize their solutions to deliver unparalleled customer experiences.

The developer experience, freedom and flexibility, cohesiveness, and interoperability are why composable architectures are the preferred choice for building marketing sites, ecommerce, consumer, and educational sites.

Architectures are not the only thing accelerating an organization's speed to market. Perhaps the most disruptive technology, since the general availability of the internet stormed onto the scene, has entered the mainstream. Welcome, Artificial Intelligence (AI).

What is the purpose of the sites you work on?



Why generative AI will make it harder for monoliths to keep up

Never in human history has there been such an explosion of available content and information as there is today. Developers, marketers, and others can now lean on Machine Learning / Artificial Intelligence (ML/AI) to create websites, content, and other digital experiences faster than ever.

The greatest web development risk to a business might surprise you—it's whether or not your existing monolithic web stacks can support the rapid pace of content creation that ML/AI is making available to organizations.

Techie friends, bear with us. Throughout this section, you'll notice we refer to machine learning, large language models, and how natural language processing accesses this information as AI. It's a mouthful, and quite repetitive, otherwise. Moving on.

AI has exploded onto the scene and we're witnessing a digital renaissance that's redefining how we think about and utilize data to perform daily tasks—i.e. work.

This is because AI can make use of trillions of data points using massive datasets to perform tasks for the user: translating, recognizing, predicting, and generating text, images, and even art from natural language.

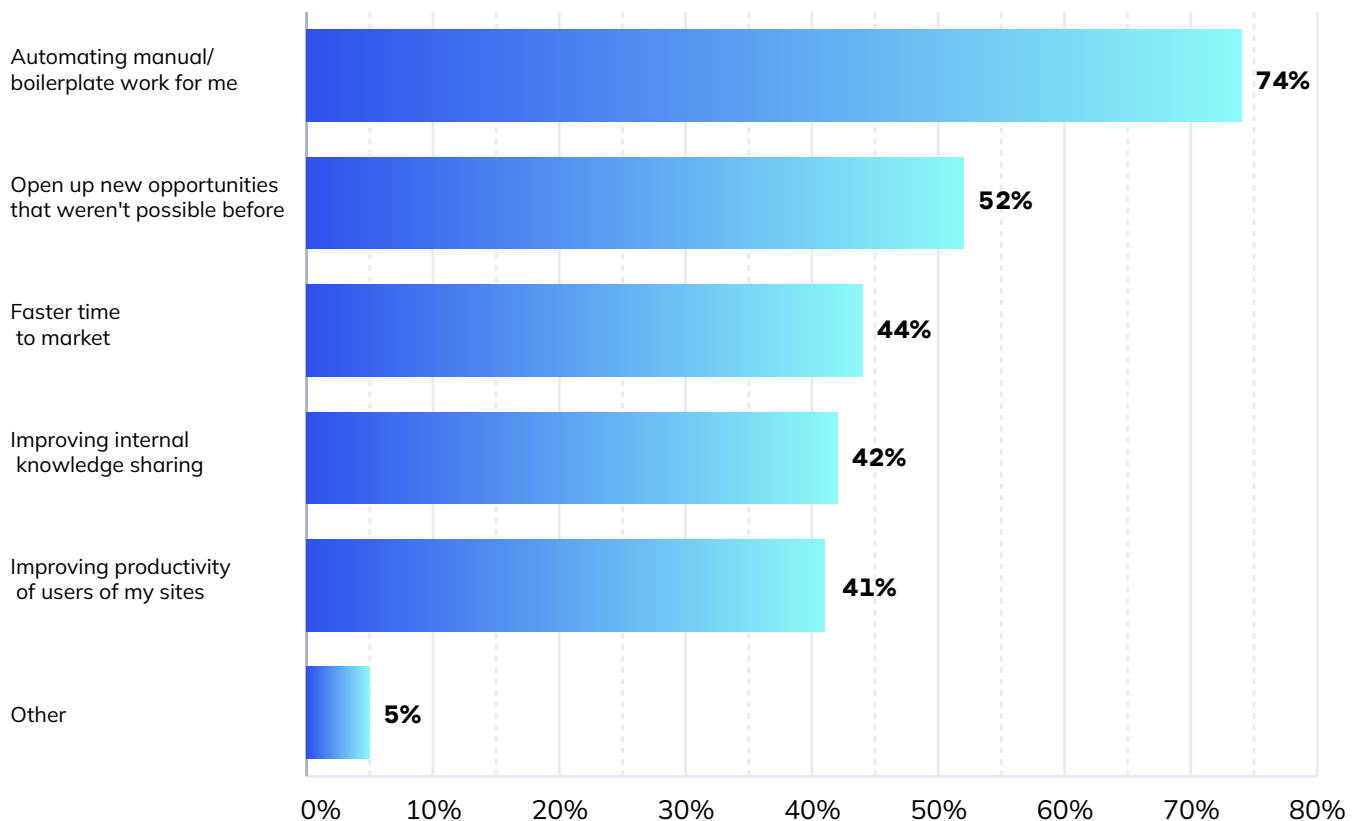
An unprecedented 80% of respondents reported using AI on the websites they're building. This explosion of usage is the highest we have ever witnessed.

Developers utilizing AI are already seeing impactful benefits of the technology:

- **Over 70%** of respondents say they are automating manual, repetitive work—meaning teams are creating more output with the same resources and while consuming less human cycles
- **More than 50%** are seeing new opportunities that weren't possible before, such as automating manual, time-intensive processes like losing hours writing glue code and reading API docs, and quickly spinning up new web projects with a simple prompt defining the framework, content management system, and hosting provider of your choosing
- **Over 40% say** they use AI to improve their internal knowledge sharing, improve productivity, and experience a faster time to market—making it possible to launch more campaigns, products, and content faster

By deploying AI solutions in their daily work, developers are realizing efficiencies in the form of time-savings and more output than they've previously seen. However, these same efficiencies are not available to everyone across the board. With legacy systems built over several years across multiple frameworks, languages, and content sources with custom, in-house solutions, the application of AI is limited. To truly embrace all benefits of AI in your daily work, the underlying technology and infrastructure matters.

In what ways do you think ML/AI can improve your workflow or the sites you build?



Pertaining to developers using AI in their workflows, we asked them to share what concerns they have. The insights were revealing:

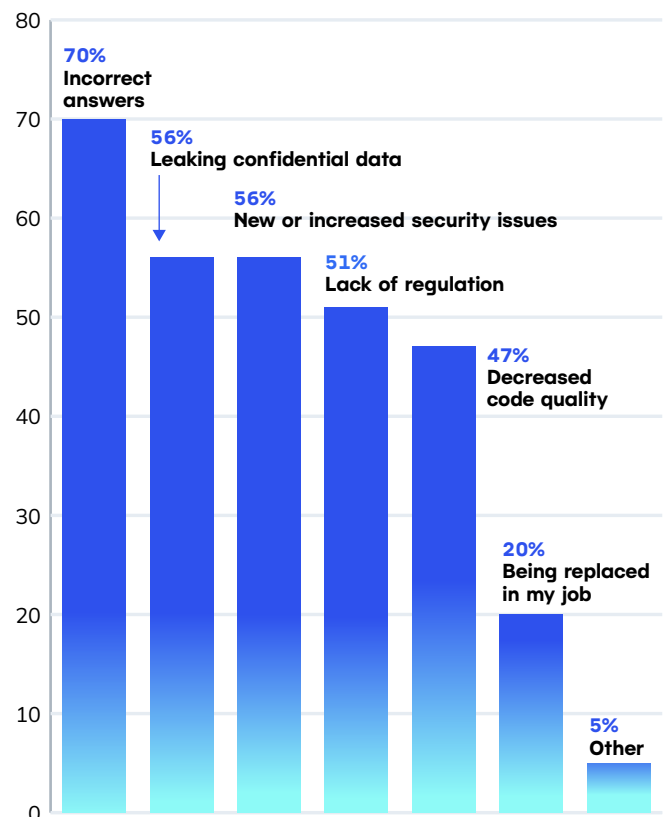
- **70%** are concerned about receiving incorrect answers and information
- **56%** think the greatest risks are new security issues and the leaking of confidential information
- **51%** say the lack of regulation is a significant barrier to the enterprise adoption of the technology
- **47%** are worried about a decrease in code quality

The risks and concerns of AI are not impacting the usage of the technology. In fact, we're confident about a few things:

1. The reported top concern—receiving incorrect answers—will be mitigated by approaches such as retrieval-augmented generation (RAG), where the model has access to a database of facts which can be integrated into the generated text.
2. Security, regulation, and risk will almost certainly be approached as security is with cloud computing. Cloud service providers had to clearly define a shared responsibility model that makes clear the difference between “security OF the cloud” and “security IN the cloud.” We foresee similar definitions regarding prompt engineering.
3. The data provided to AI tools and operational security measures will be managed through policies enforced by the organizations.

In summary, AI is defining a new velocity for bringing web projects to market. This redefined pace of launching campaigns, websites, and content makes one thing clear: organizations will need the right architectures and workflows in place to support it. Businesses using monoliths for their revenue-driving web properties will likely be outpaced by organizations utilizing composable architectures and AI.

What risks do you think are posed by the adoption of ML/AI tools?

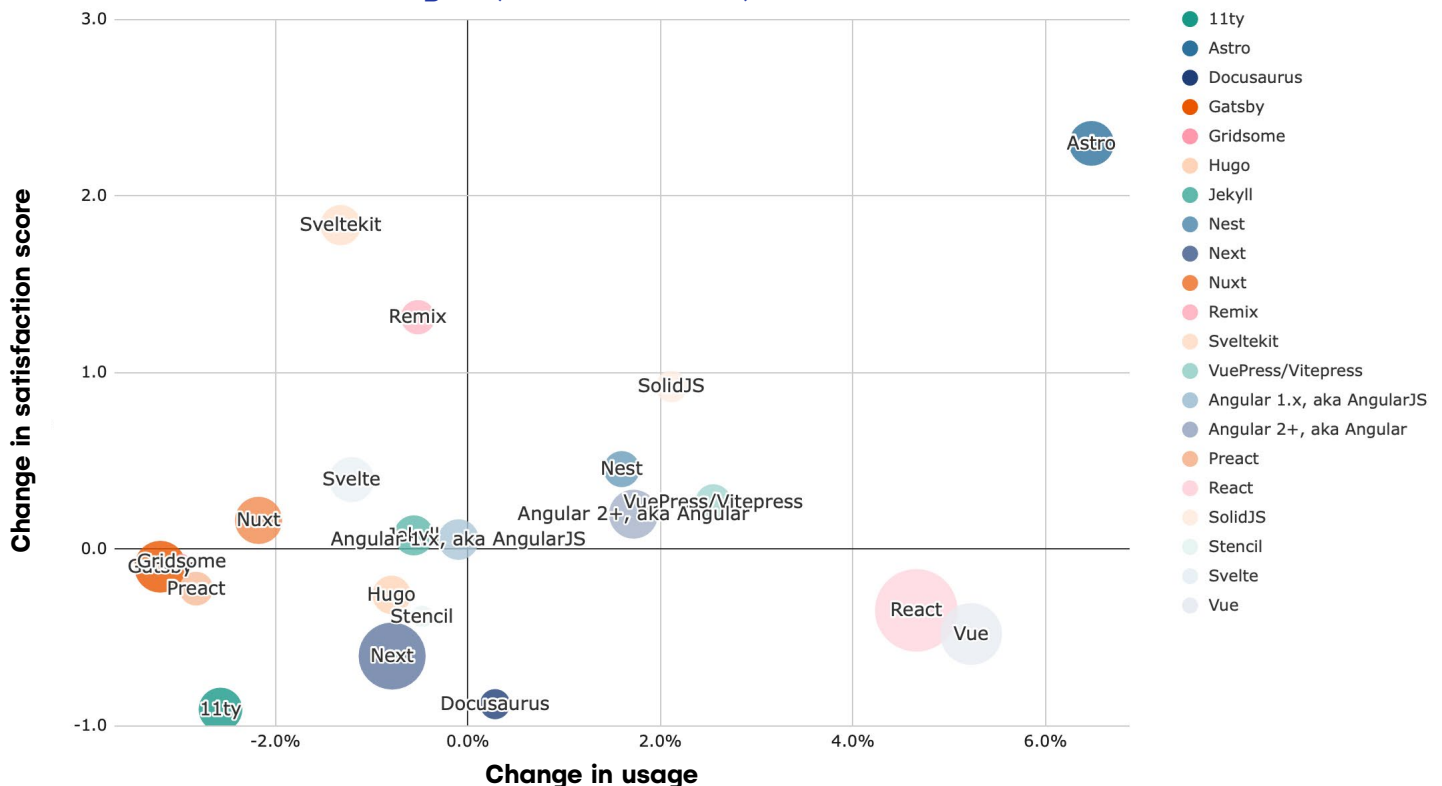


What is the top-rising framework in satisfaction from 2022 to 2023?

In the last two years, we reported Next.js as the top framework in both usage and satisfaction. While Next.js is the dominant framework for projects used in the last 12 months and the overall satisfaction with the framework is positive, Astro is showing the highest growth in both usage and satisfaction. It tops the leaderboard in growth by a wide margin, while maintaining one of the strongest growth numbers in satisfaction.

What does this mean? From previous surveys and years of data, maintaining satisfaction while expanding usage is the top indicator for expected future growth. We can conclude that we are only seeing the beginning of what Astro will do in the future. Previously Next.js has held great scores on both measures, but has taken a sharp turnaround this year. Next usage in 2022 was a staggering 47% with a satisfaction rate of 4.2 compared to 2023's results showing a 46.2% usage paired with a steep drop of satisfaction to 3.6. Astro on the other hand nearly doubled in usage and increased in satisfaction to an impressive 6.8 in 2023 from 4.5 in 2022.

Rendering framework or site generator usage and satisfaction changes (2023 vs 2022)



It's important to note that the data we are referencing is not just for personal blogs and sites. Some developers are building sites that have monthly traffic in the hundreds of thousands and others in the millions.

The trick is not to be widely popular with a very niche audience to score well with survey respondents, but for developers to remain satisfied with the product as it grows in features and popularity in parallel.

Here are three reasons we think Astro is accelerating in both usage and satisfaction within the developer community:

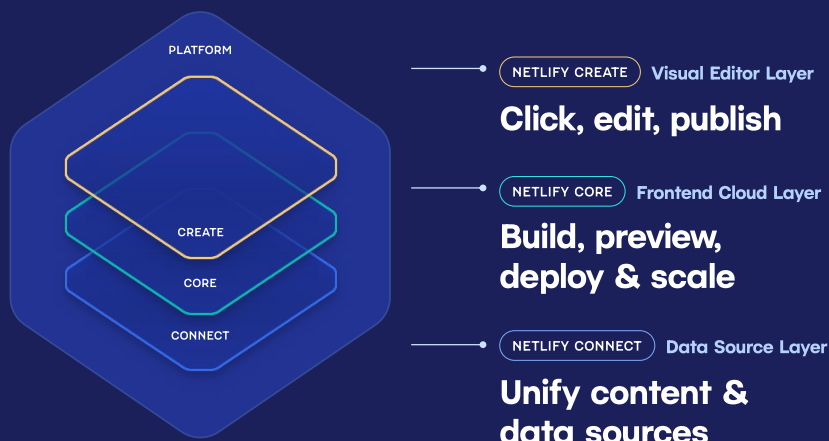
- 1. It's easy:** Astro has done an incredible job in becoming an “easy place” to begin building. We think Astro is incredible and easy due to:
 - Great CLI
 - Really good, usable templates
 - Fantastic documentation for beginners and experienced developers
 - Astro is not overwhelming. As some frame works grow over time, they tend to become more and more complicated. Astro has done an admirable job of keeping things simple while adding new features and enhancements
- 2. UI-agnostic:** or as Astro says, “Bring Your Own UI Framework” (BYOF). Astro is compatible with React, Preact, Solid, Svelte, Vue, and Lit, as well as plain HTML.
- 3. Astro Community:** Few places on the internet tout a more helpful community than Astro. With over 10,000 active developers on the community Discord, you can easily find support and make the most out of building with the help of thousands of other happy developers.

In summary, the modern web development space is moving and changing faster than ever due to incredible advancements in technology, a growing ecosystem of solutions, and the emergence of the next evolution of development. We're excited to bring you these results and want to thank everyone who participated in this year's report. Your insights and willingness to share your experiences, ideas, and thoughts are critical to building the best future of the web possible.

Making the switch to a composable architecture may be daunting. However, Netlify makes it easy to incrementally migrate your website without any downtime...and we provide you with a place to start wherever you and your organization are at in your composable journey.

The Netlify Composable Web Platform offers three solutions in a single platform that provide your teams with streamlined orchestration from all data and content sources, collaboration and execution within a single workflow, and the ability to ship website updates faster than ever before.

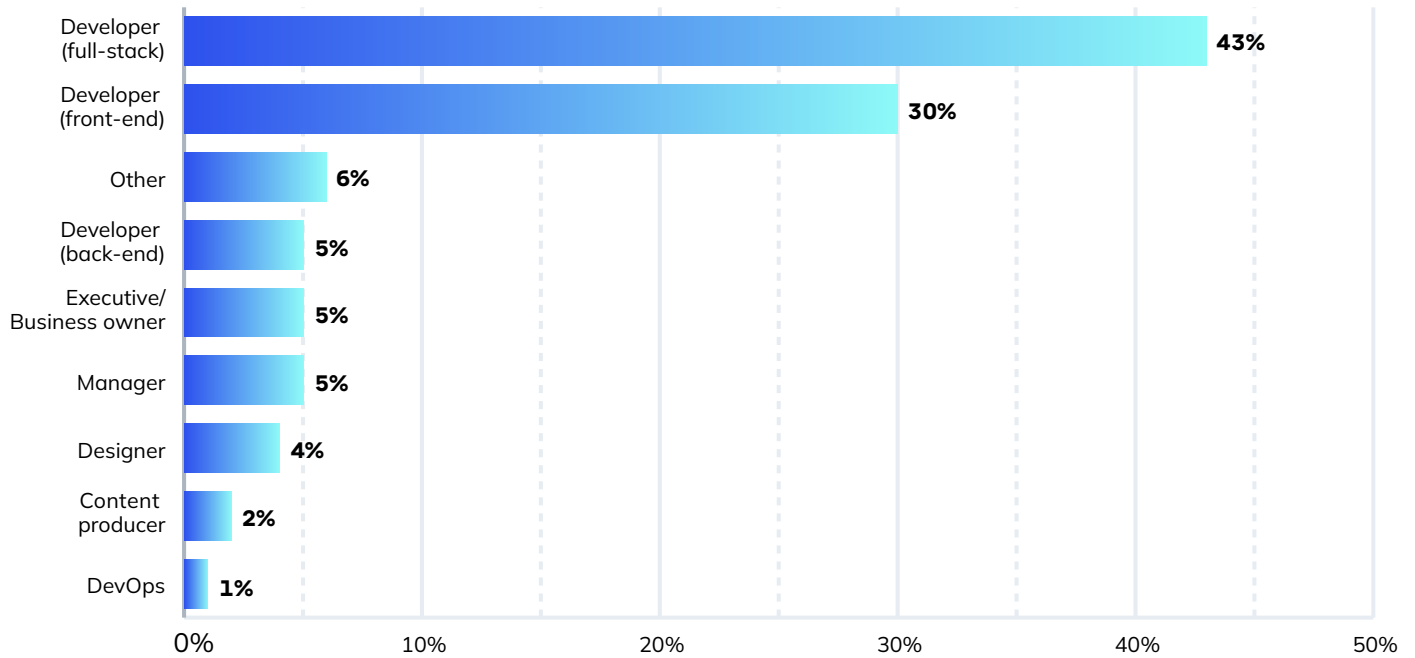
Compatible with all modern frameworks, including Astro, Next.js, and more, and flexible to connect with any legacy or modern content source, Netlify is building solutions to meet the needs of modern web developers and organizations of all sizes.



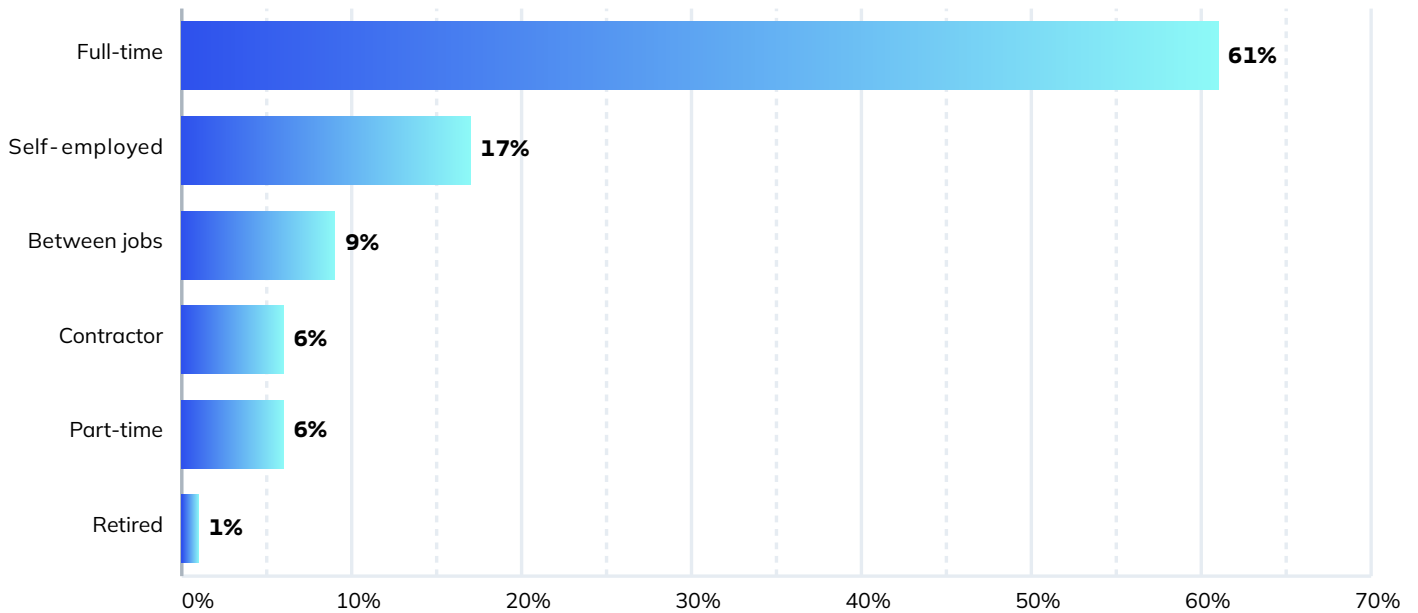
To learn more about Netlify and its solutions, visit <http://www.netlify.com>. We'd be happy to speak with you about your goals, the goals of your organization, and how modern, composable architectures can help you do more and deliver exceptional results for your end customers.

Let's Connect

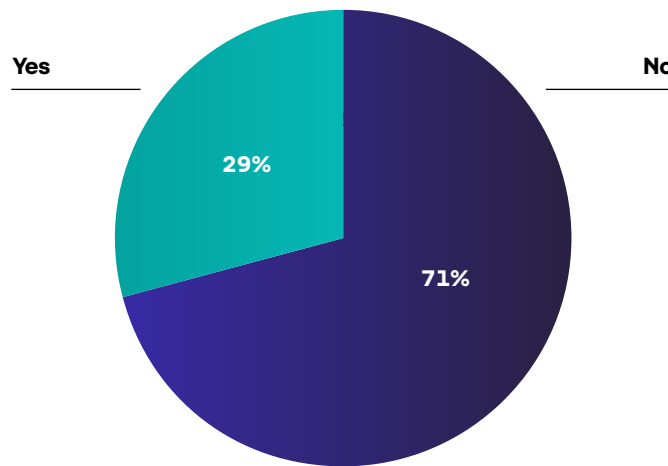
Job Title



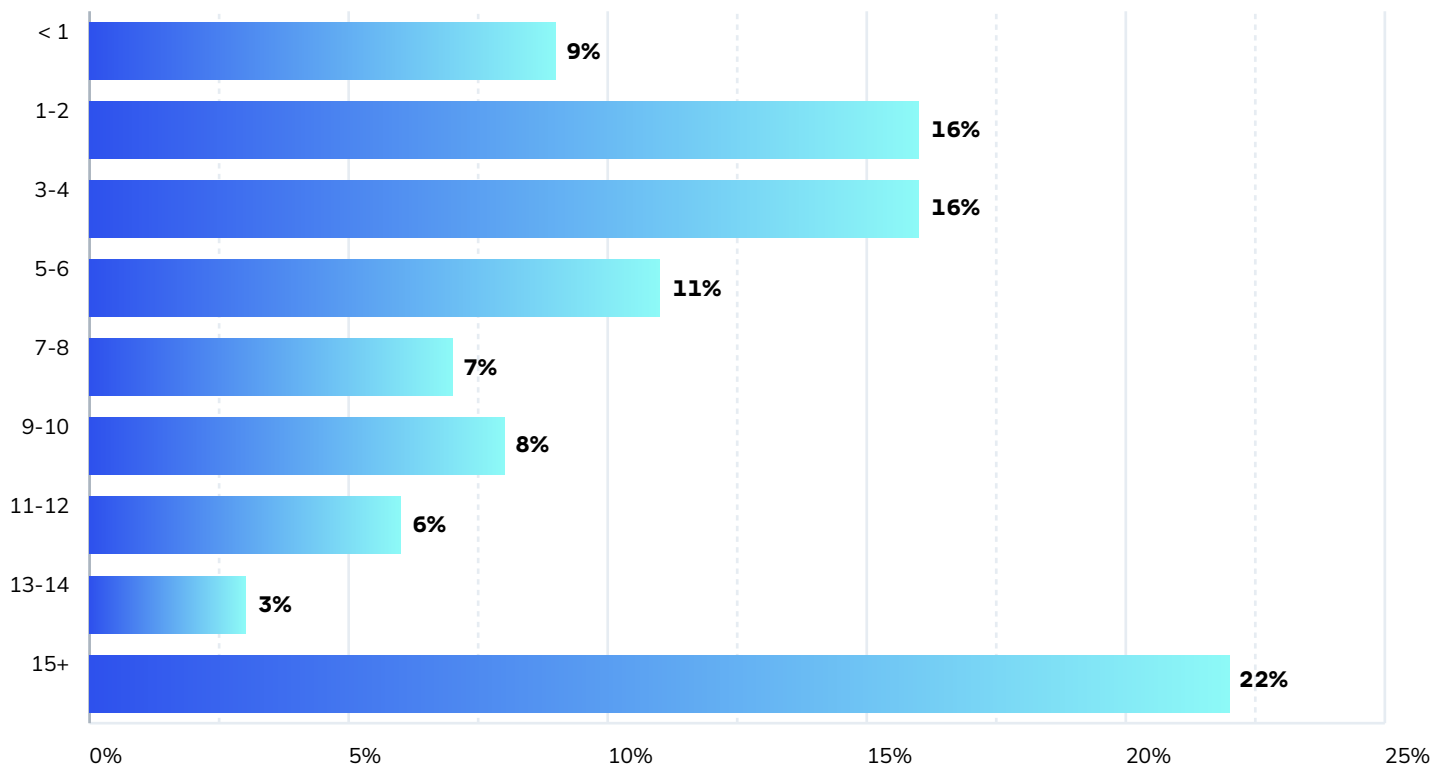
Employment Status



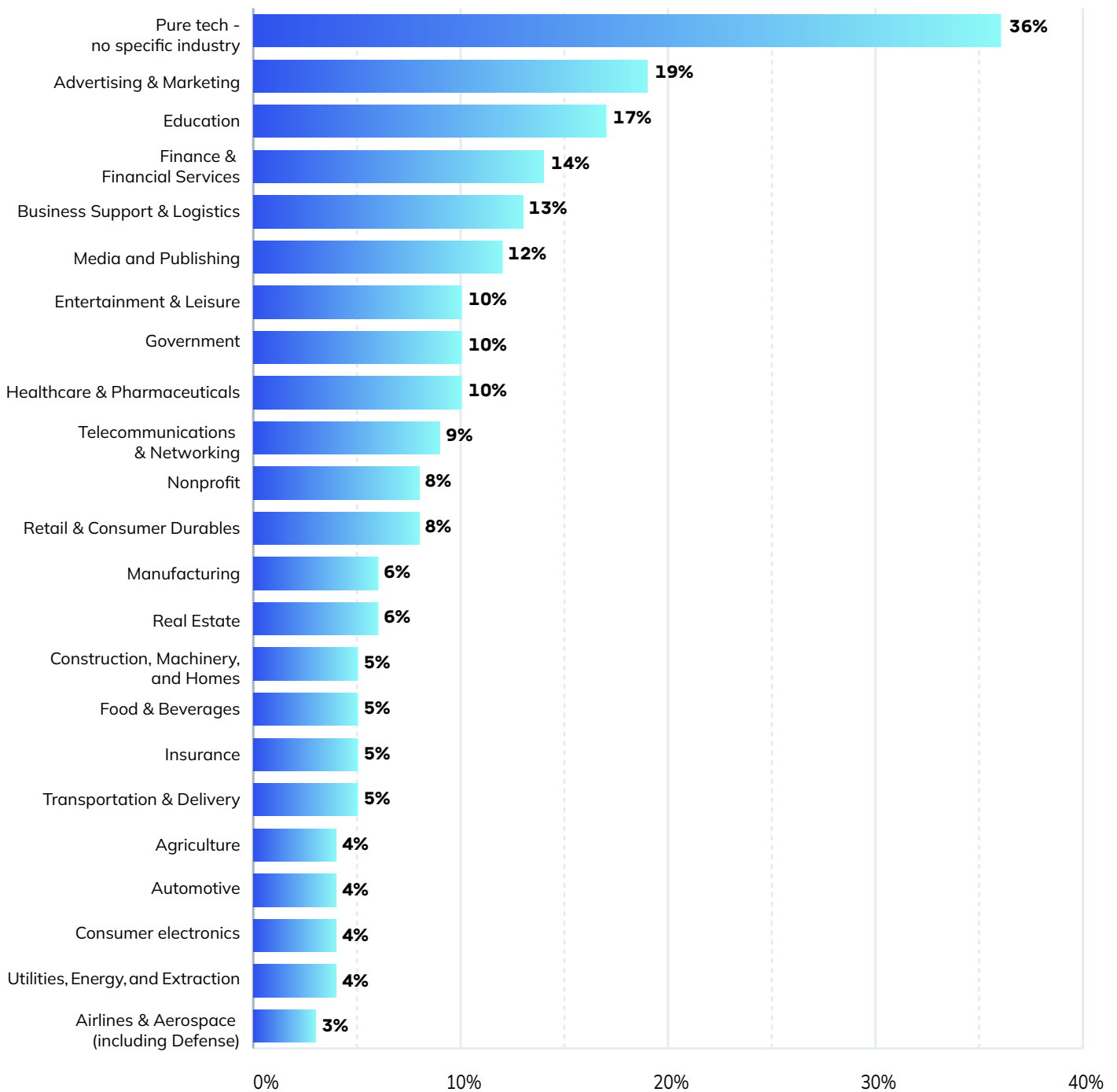
Do you work at an agency?



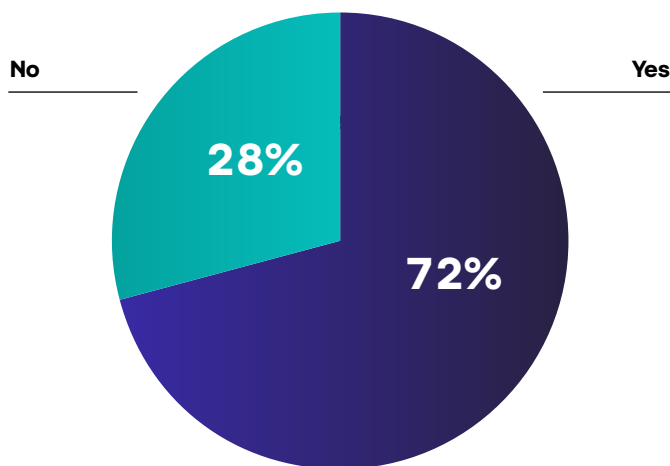
Years of Experience



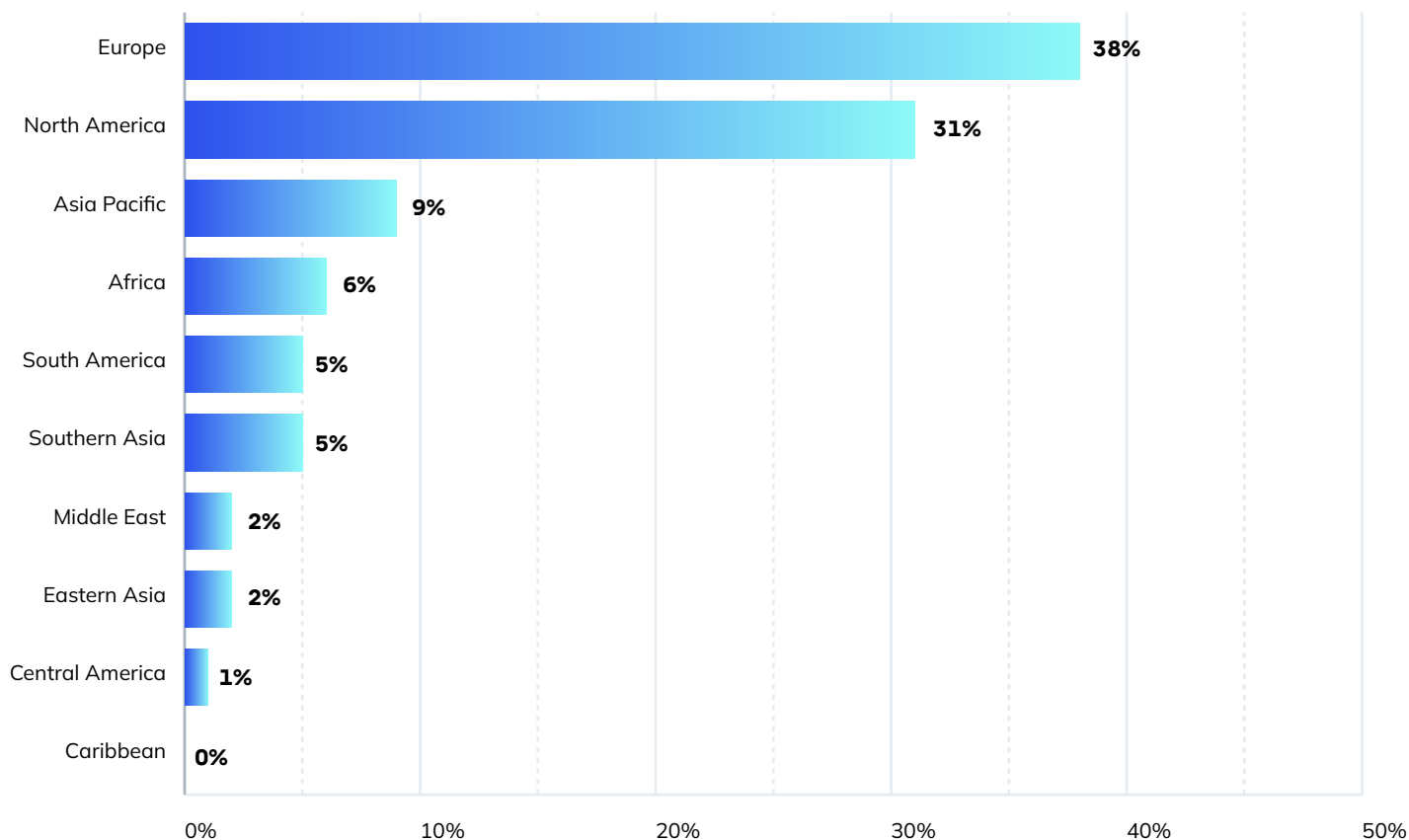
What industry is the work you do part of?



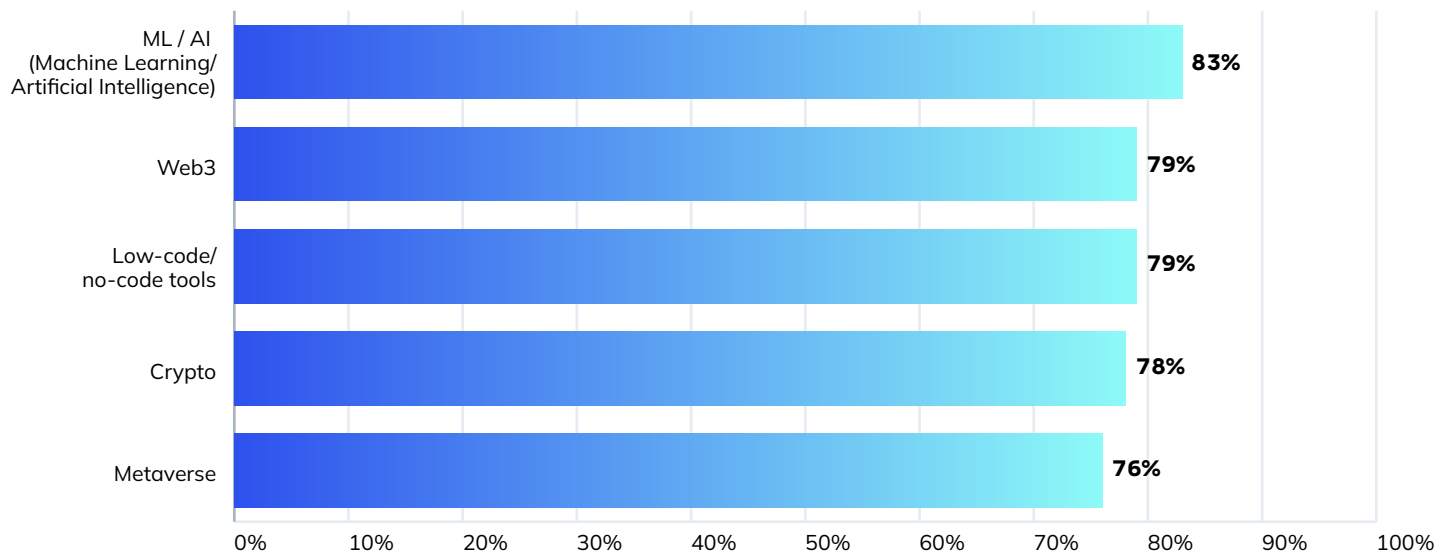
Do you consider the company you work at to be a "tech" company?



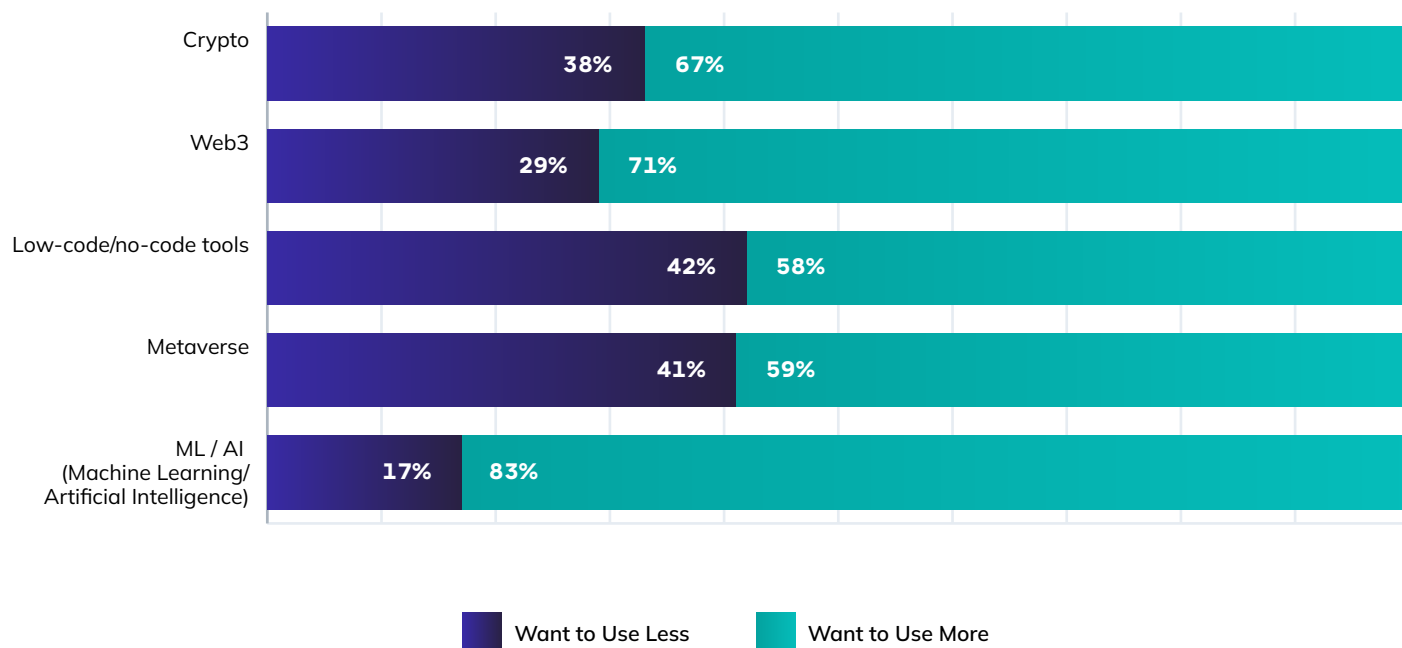
What region of the world is the best description of where you live?



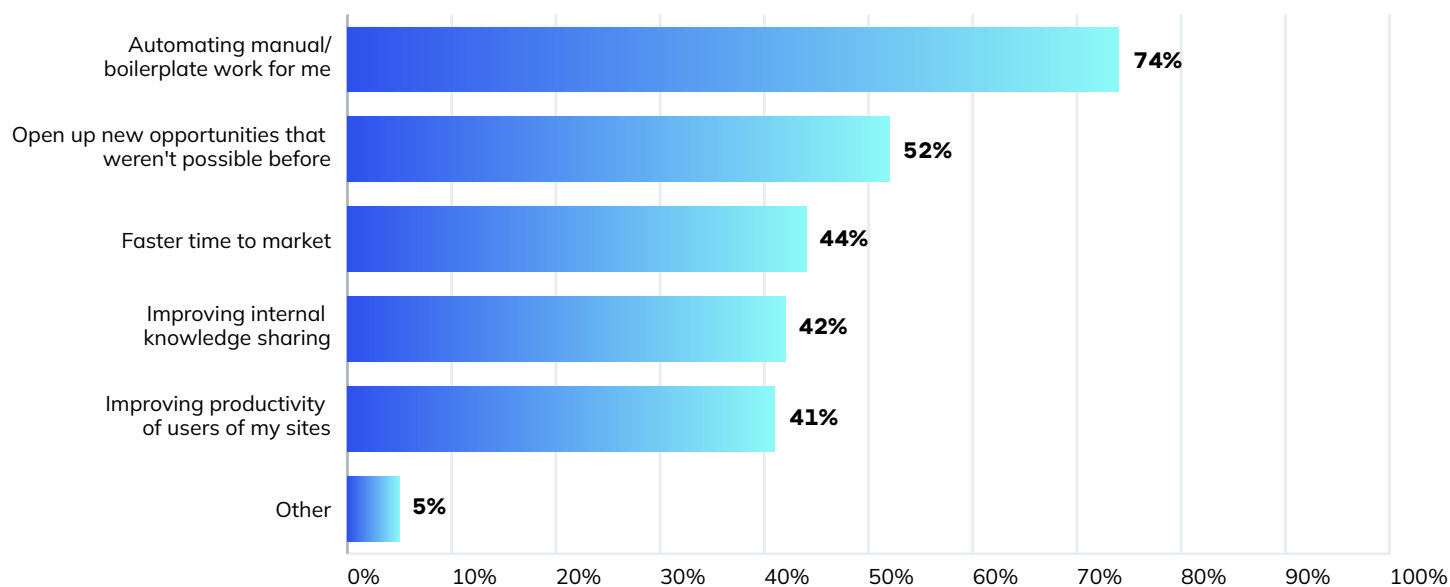
How much have the sites you've worked on in the last 12 months used any of the technologies below?



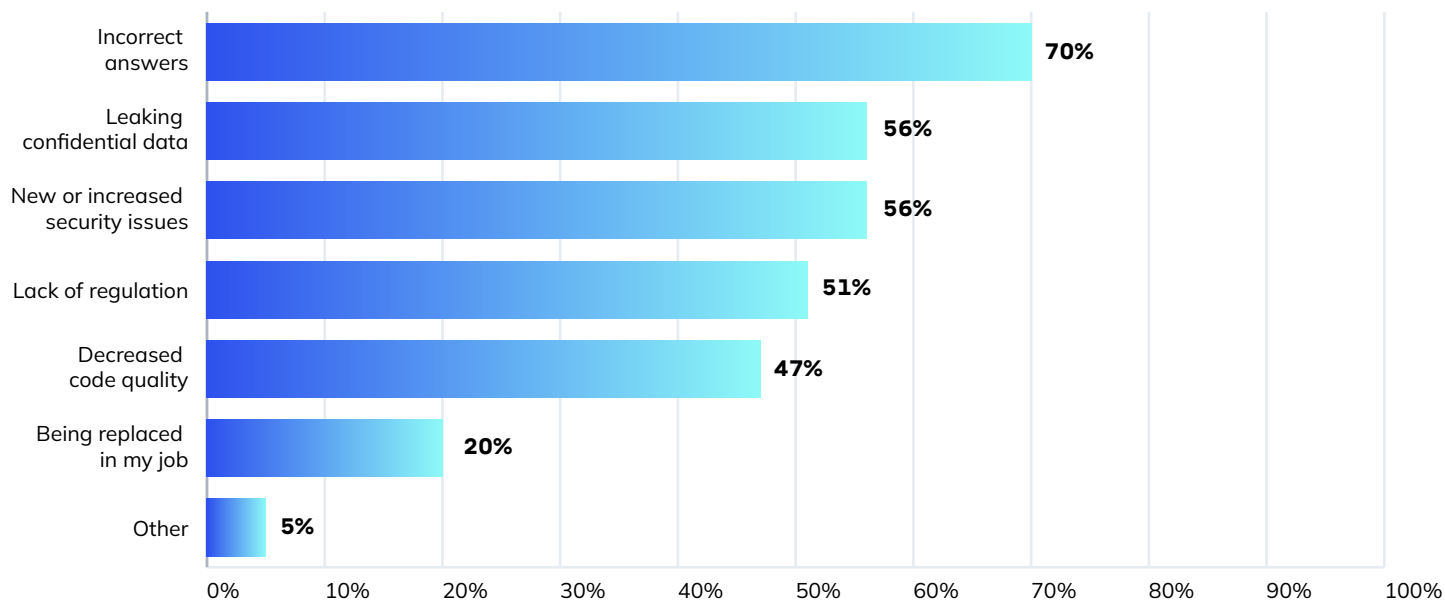
Continuing Interest in Using Modern Technologies



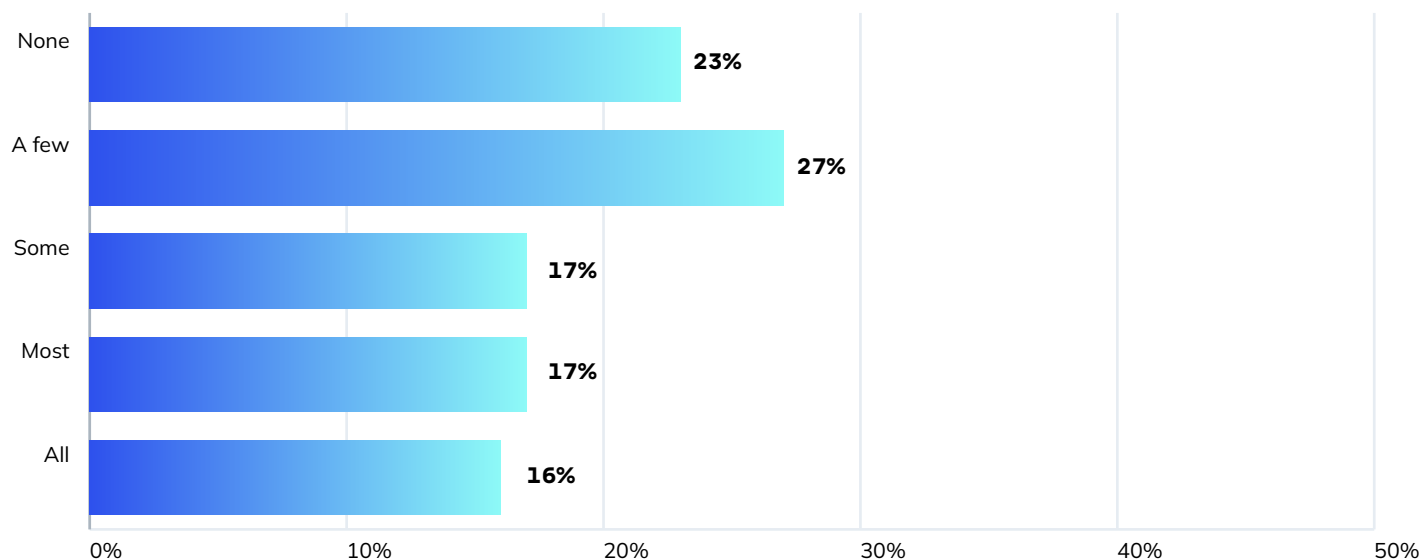
In what ways do you think ML/AI can improve your workflow or the sites you build?



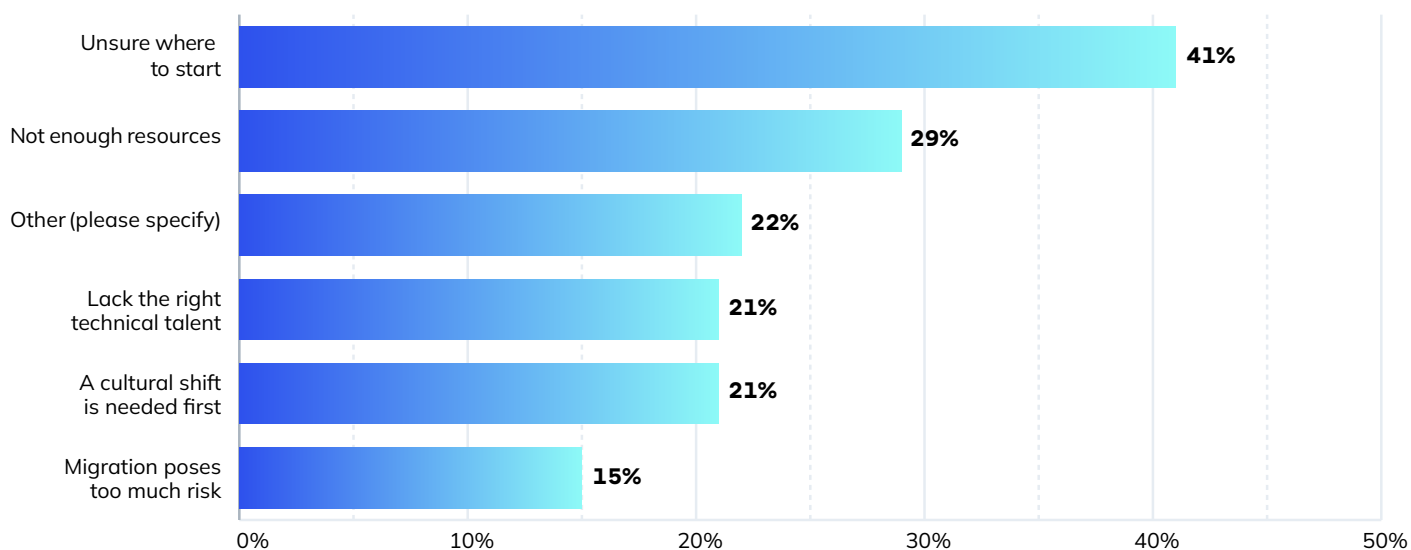
What risks do you think are posed by the adoption of ML/AI tools?



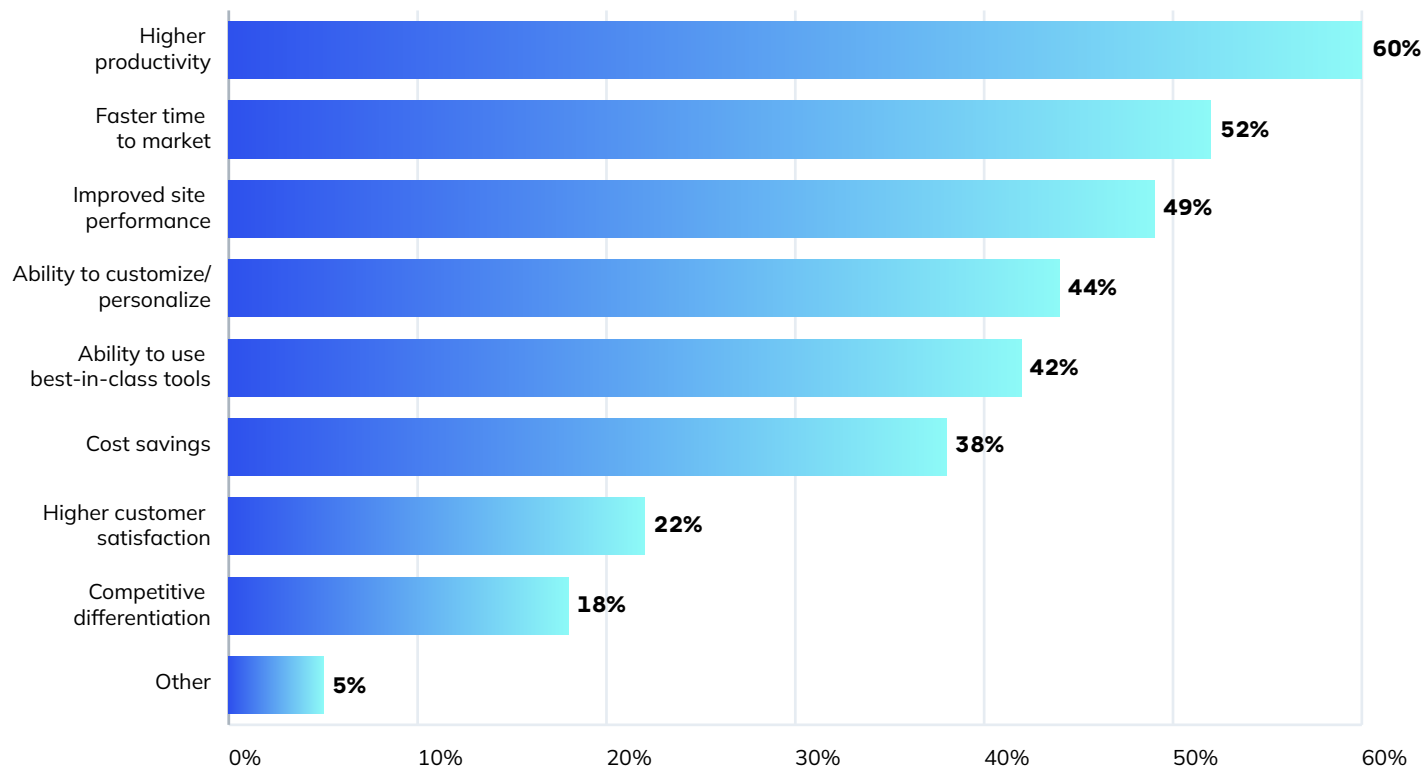
How many of the sites you've built in the last 12 months were composable using the definition above?



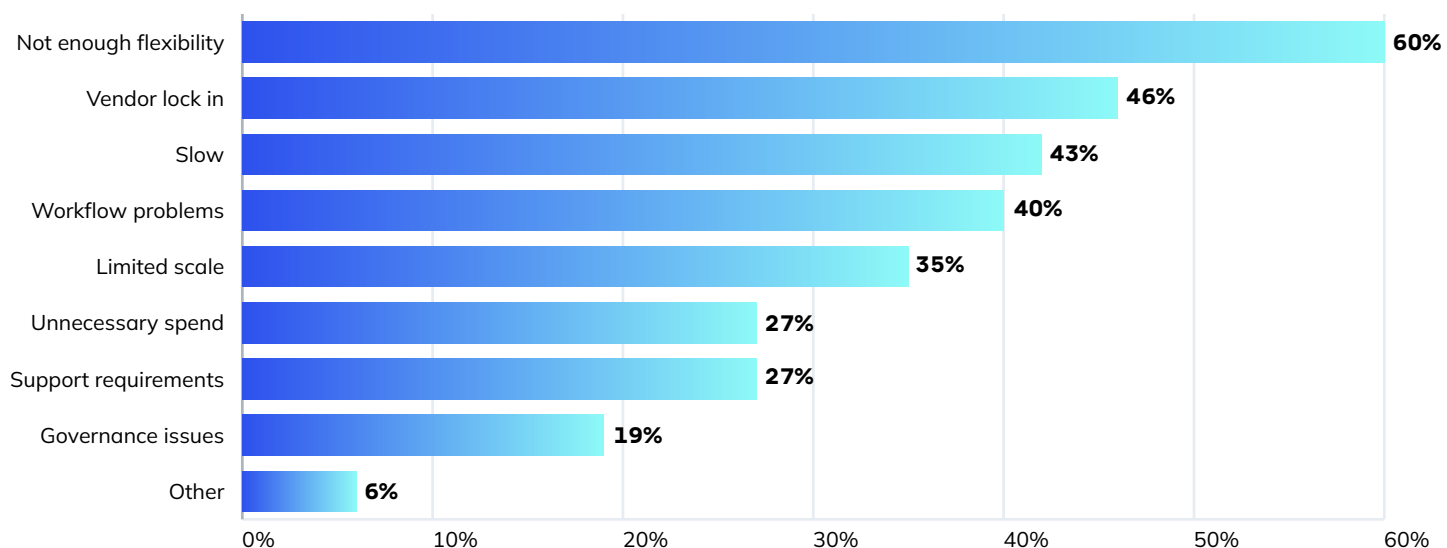
What barriers have prevented you switching to composable sites?



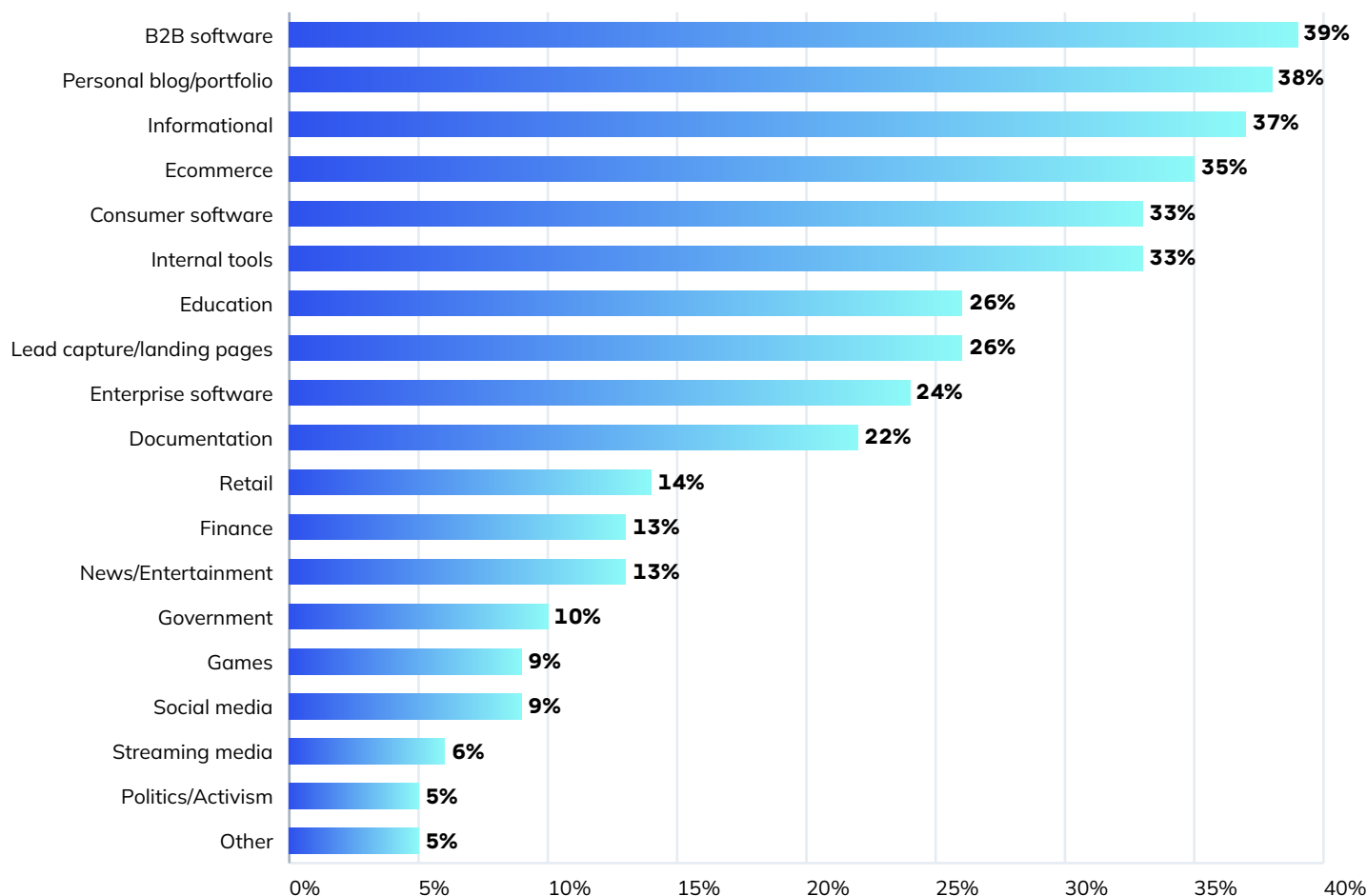
What benefits have you seen when building composable sites vs. legacy architectures?



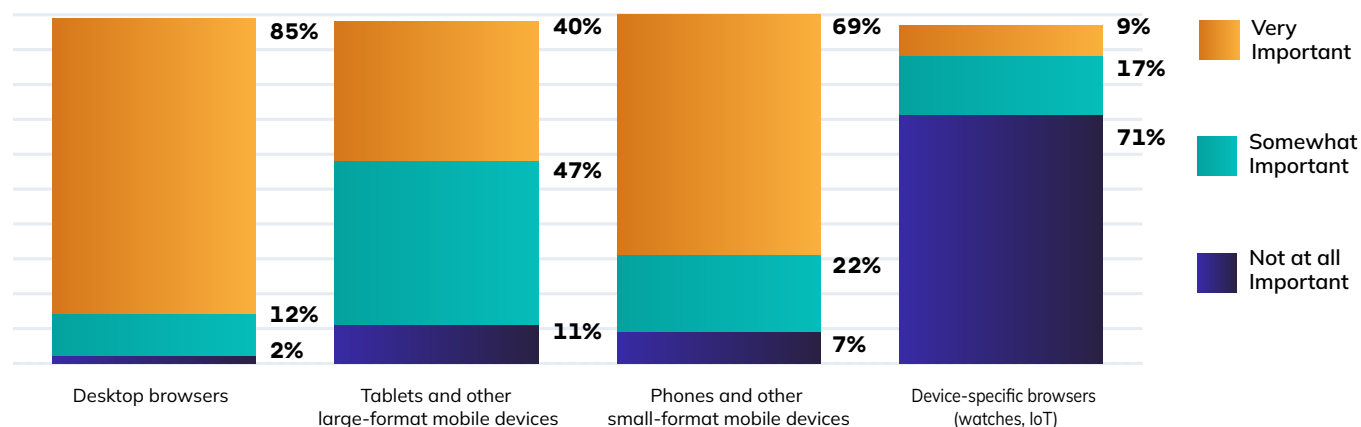
What would you say are the limitations of these monolithic platforms?



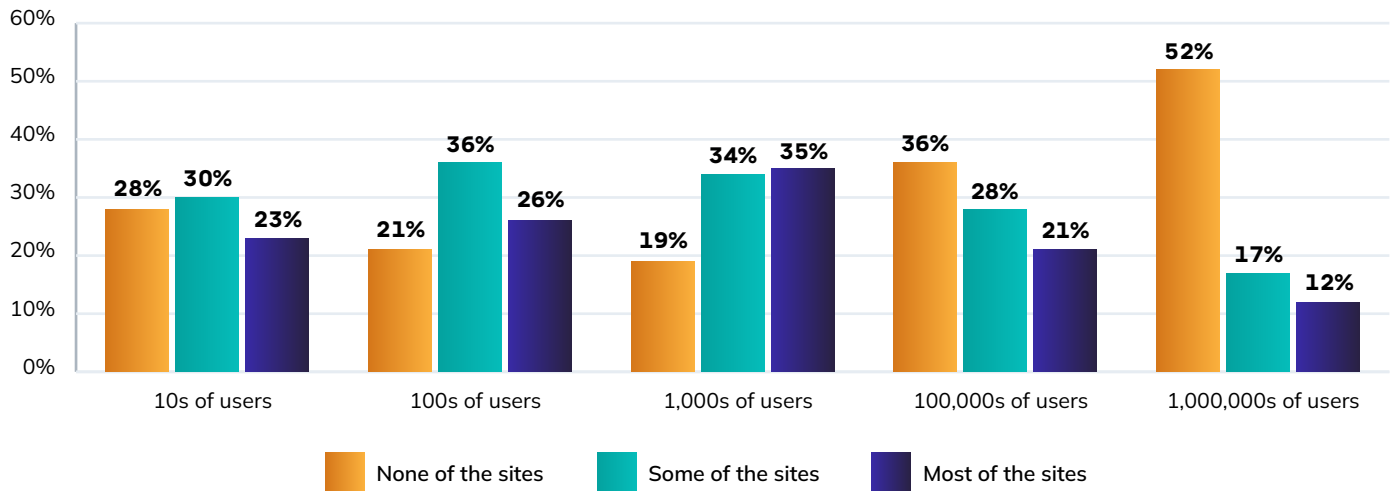
What is the purpose of the sites you work on?



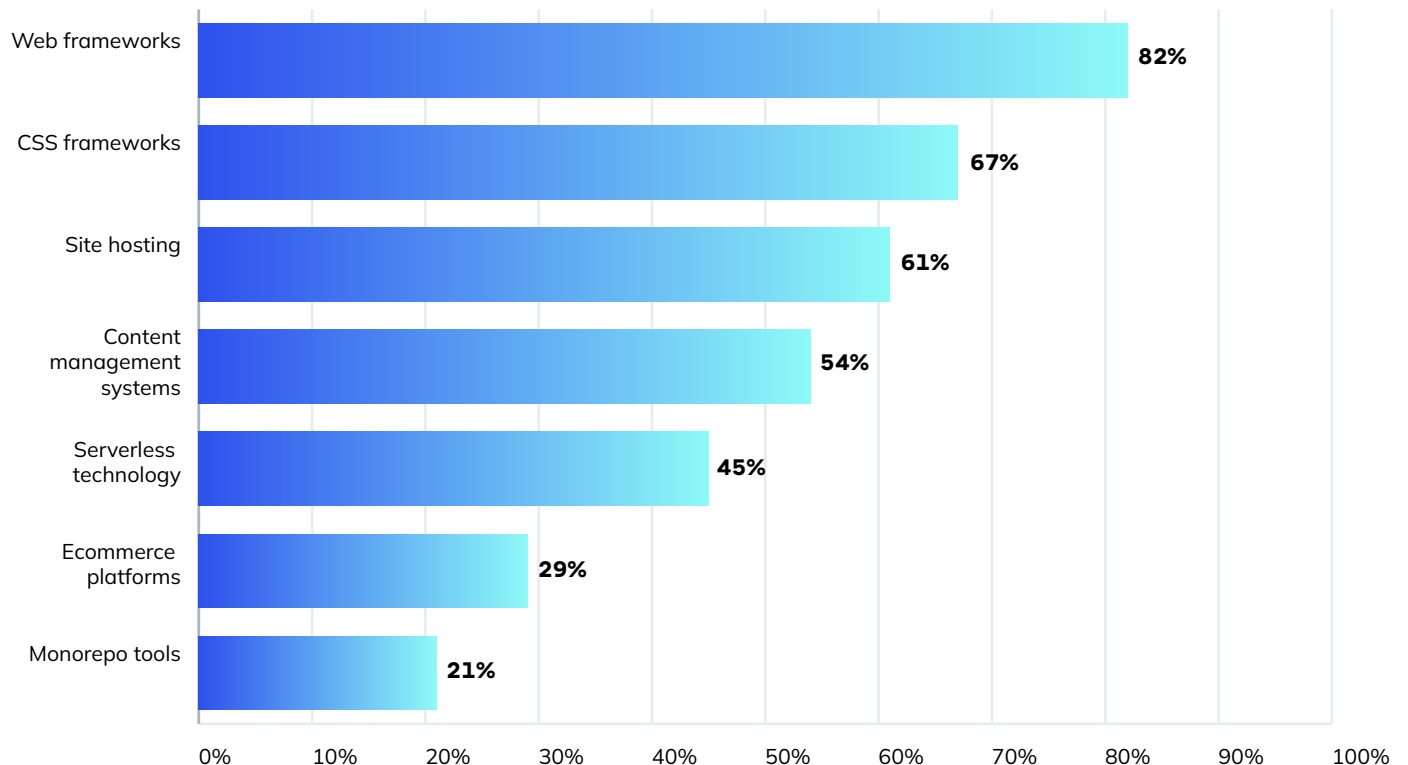
How important are users in the following environments for the sites you work on?



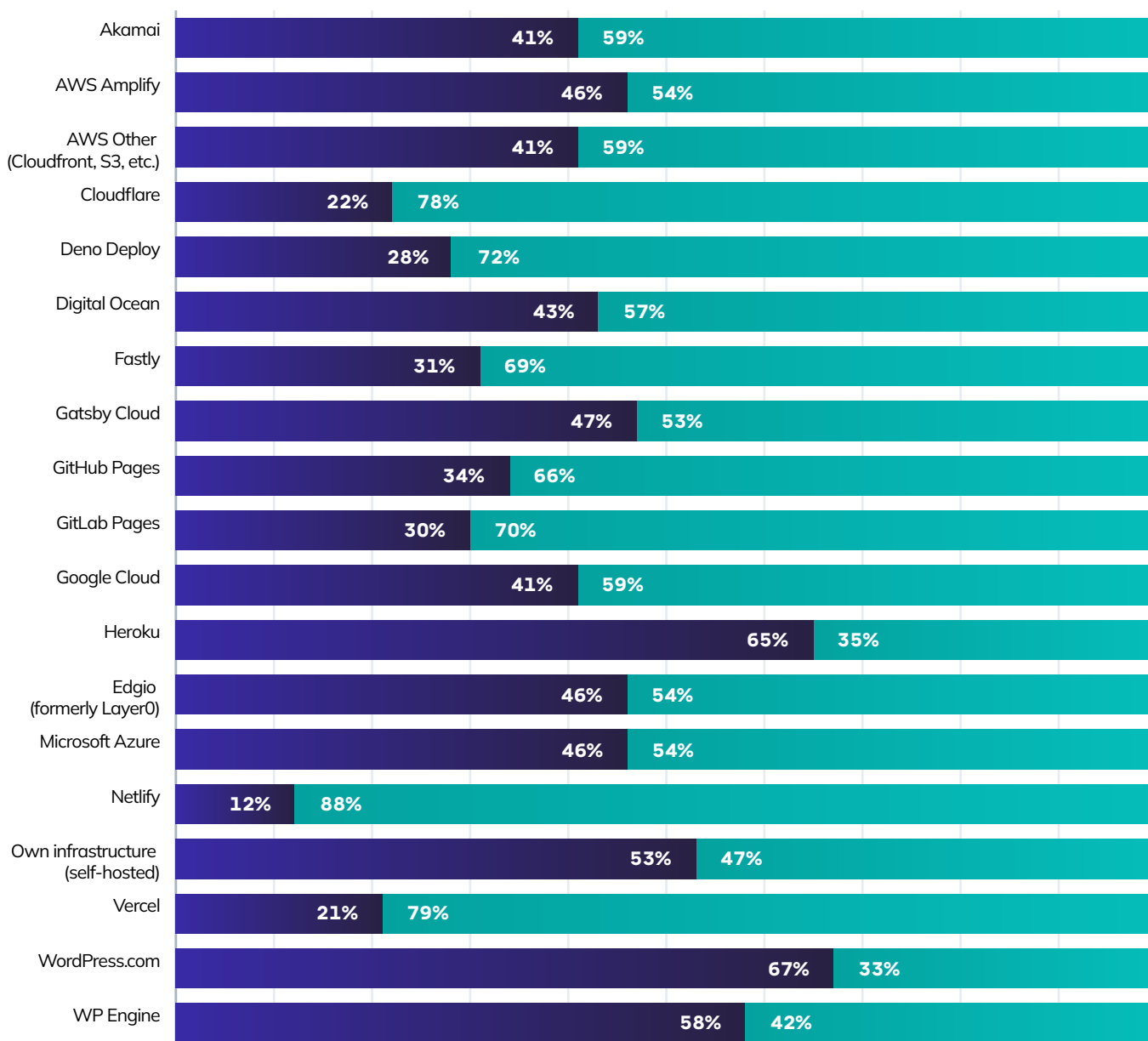
Think of the sites you've worked with in the past 12 months, how many users were they intended to serve?



Which of these aspects of modern websites do you have opinions on?

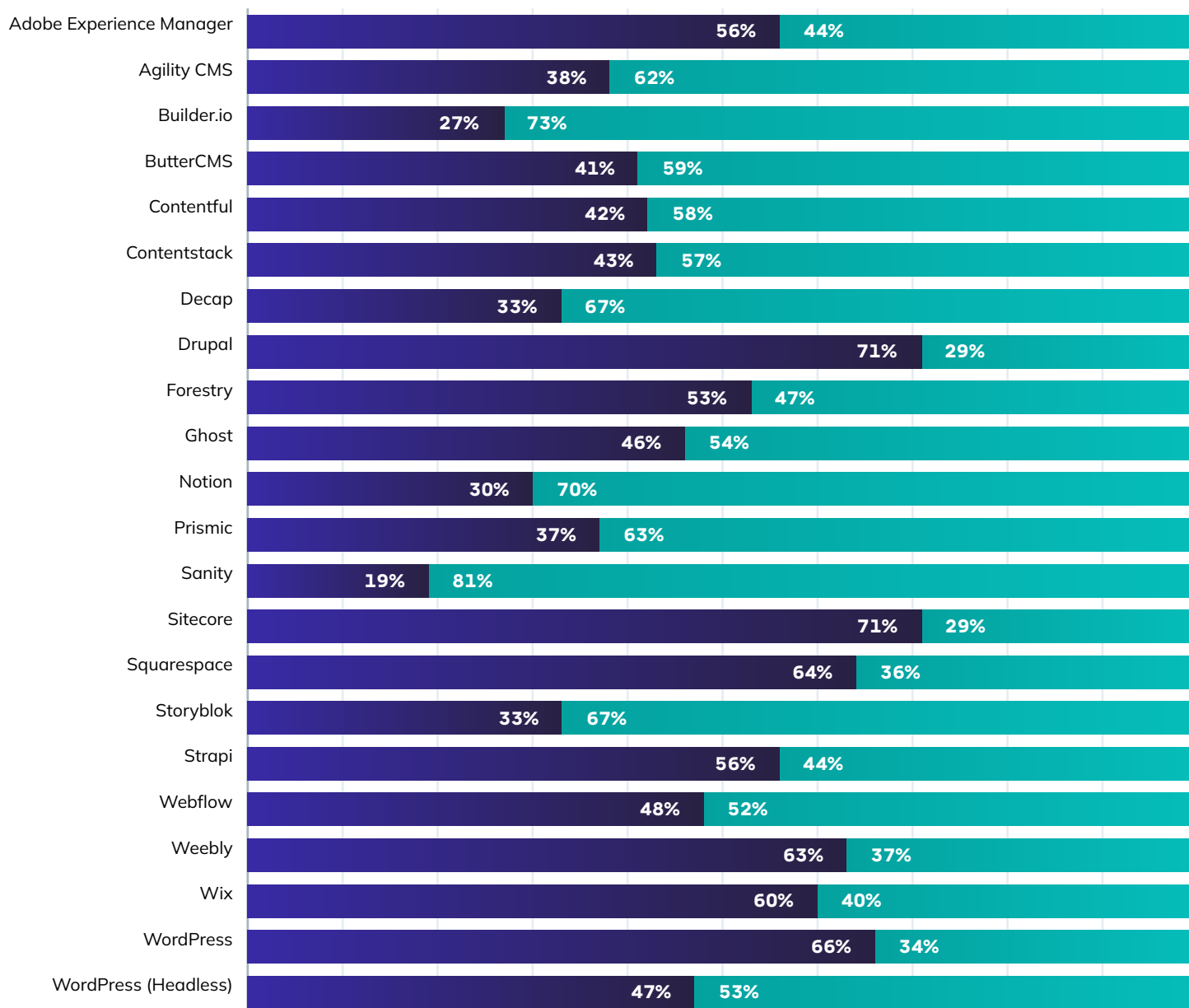


In the last 12 months, have you used any of the following services, and do you want to continue using them?



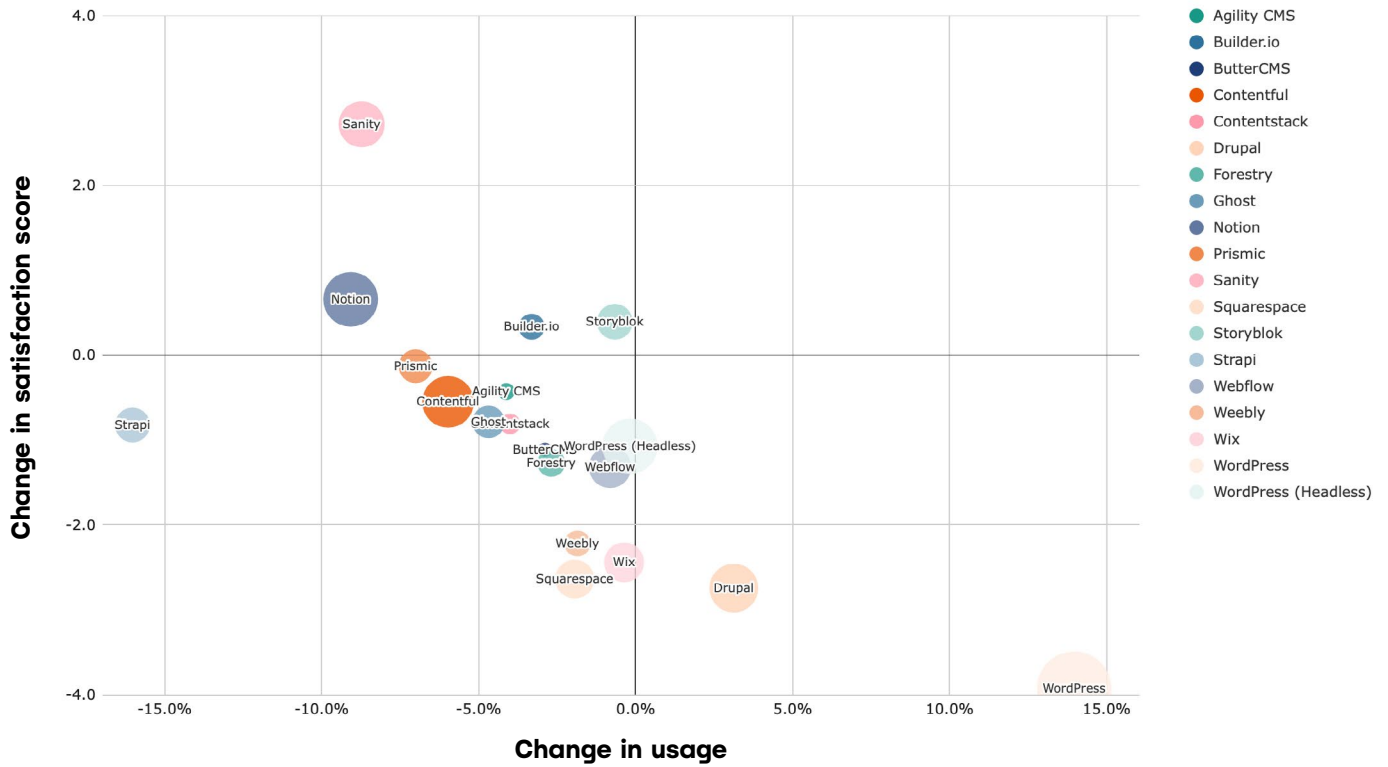
Want to Use Less
 Want to Use More

In the last 12 months, have you used any of these tools, and do you want to continue using them?

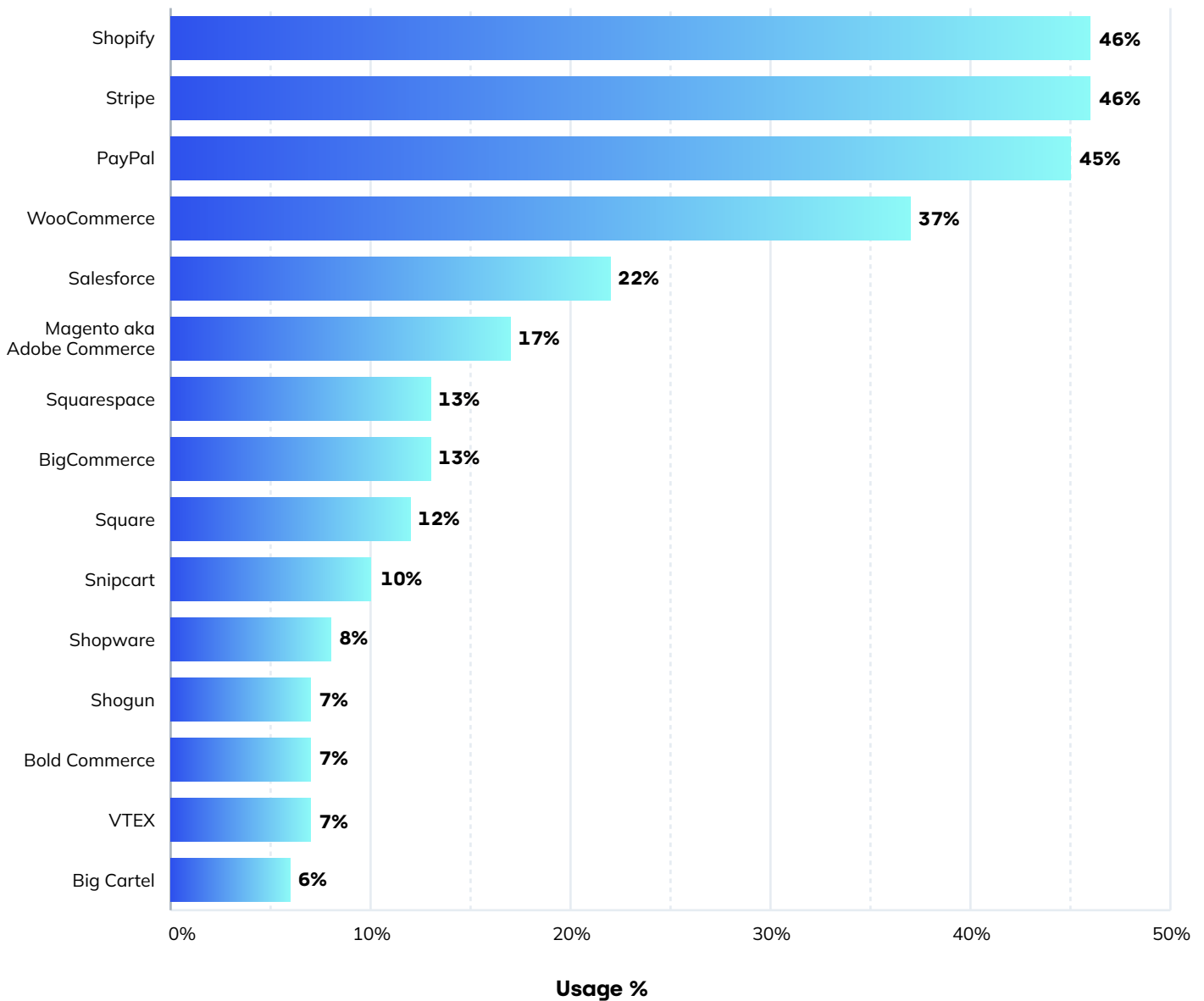


Want to Use Less
 Want to Use More

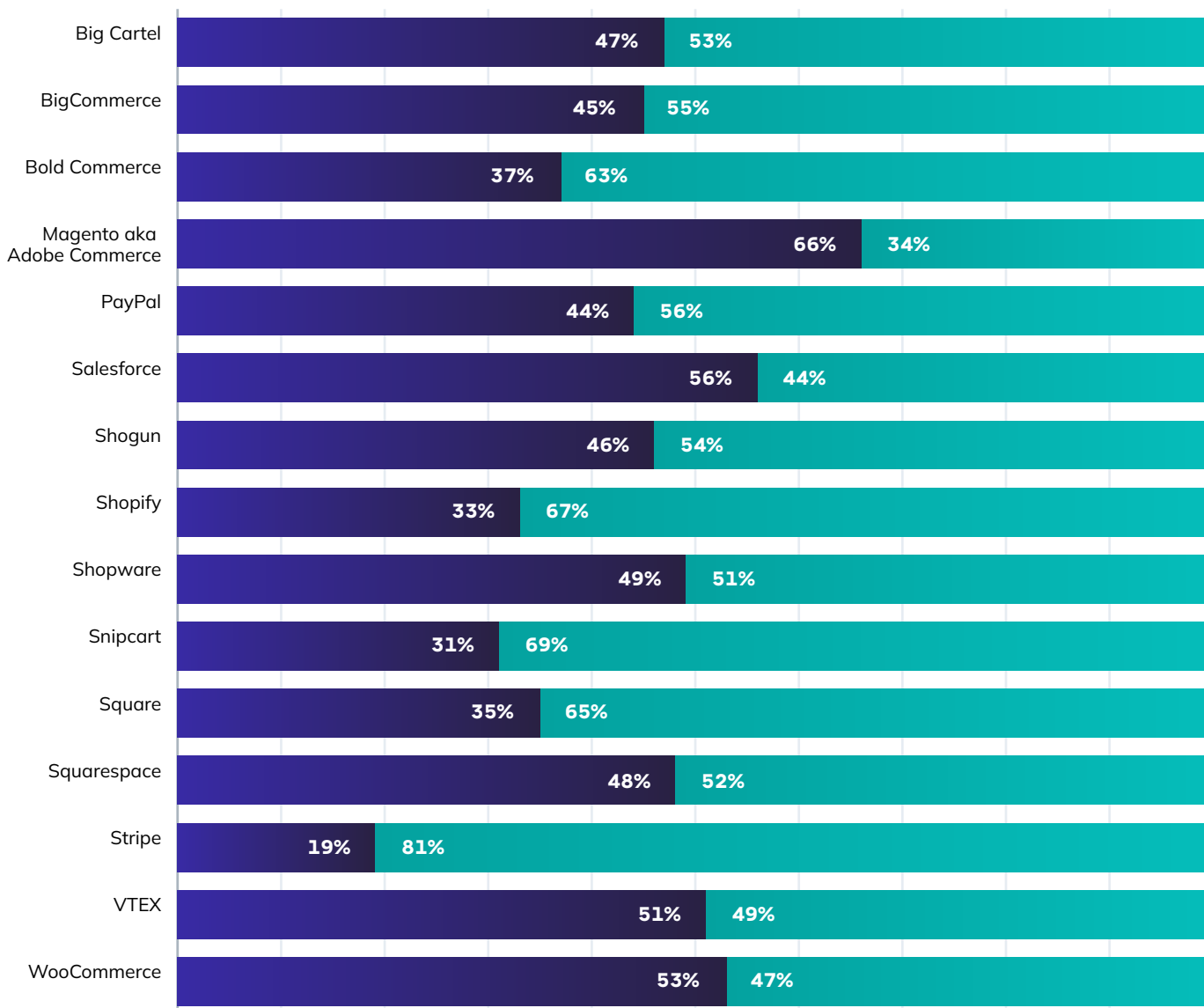
CMS usage and satisfaction changes (2023 vs 2022)



In the last 12 months, which of these ecommerce platforms have you used and do you want to continue using them?

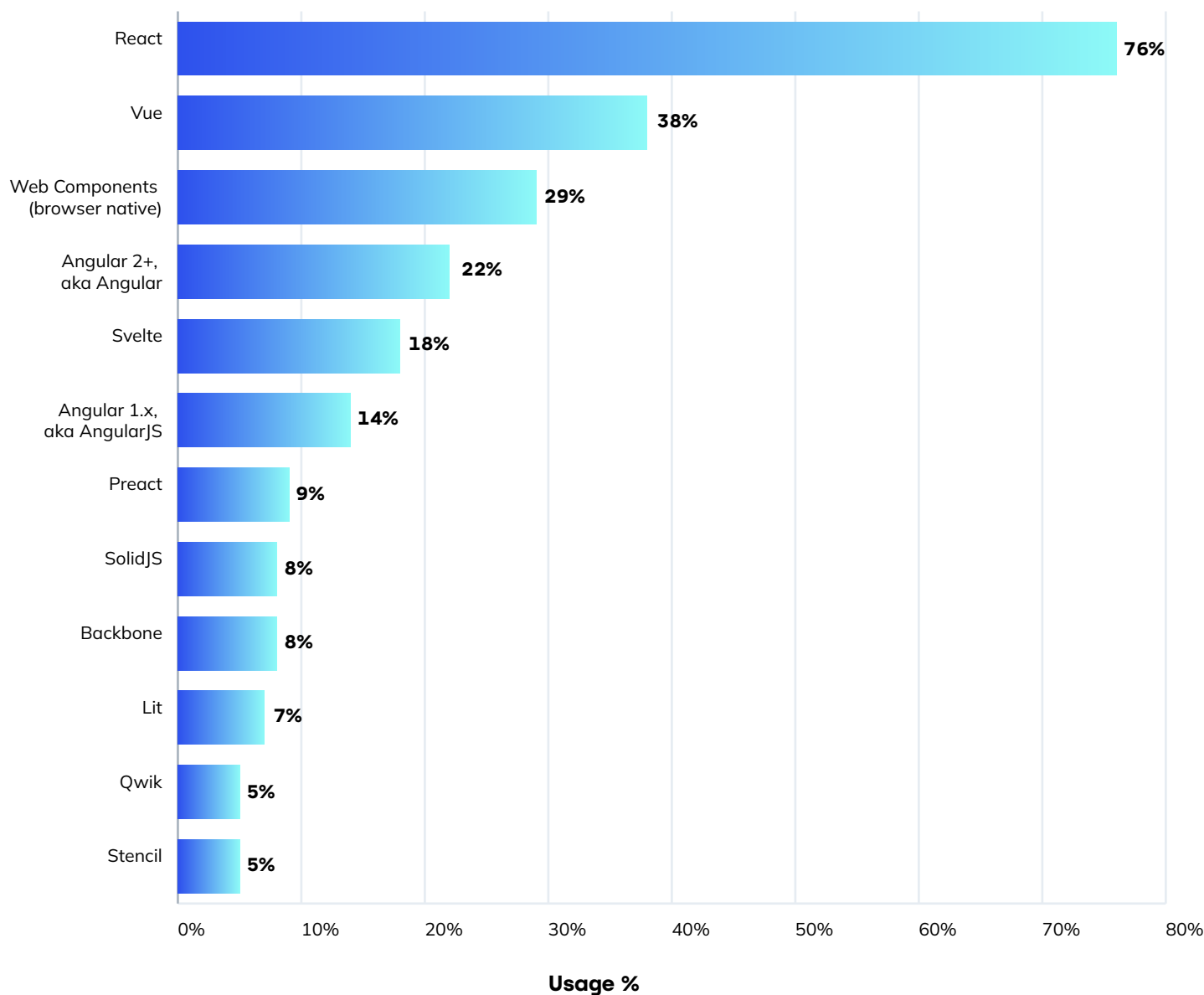


In the last 12 months, which of these ecommerce platforms have you used and do you want to continue using them?

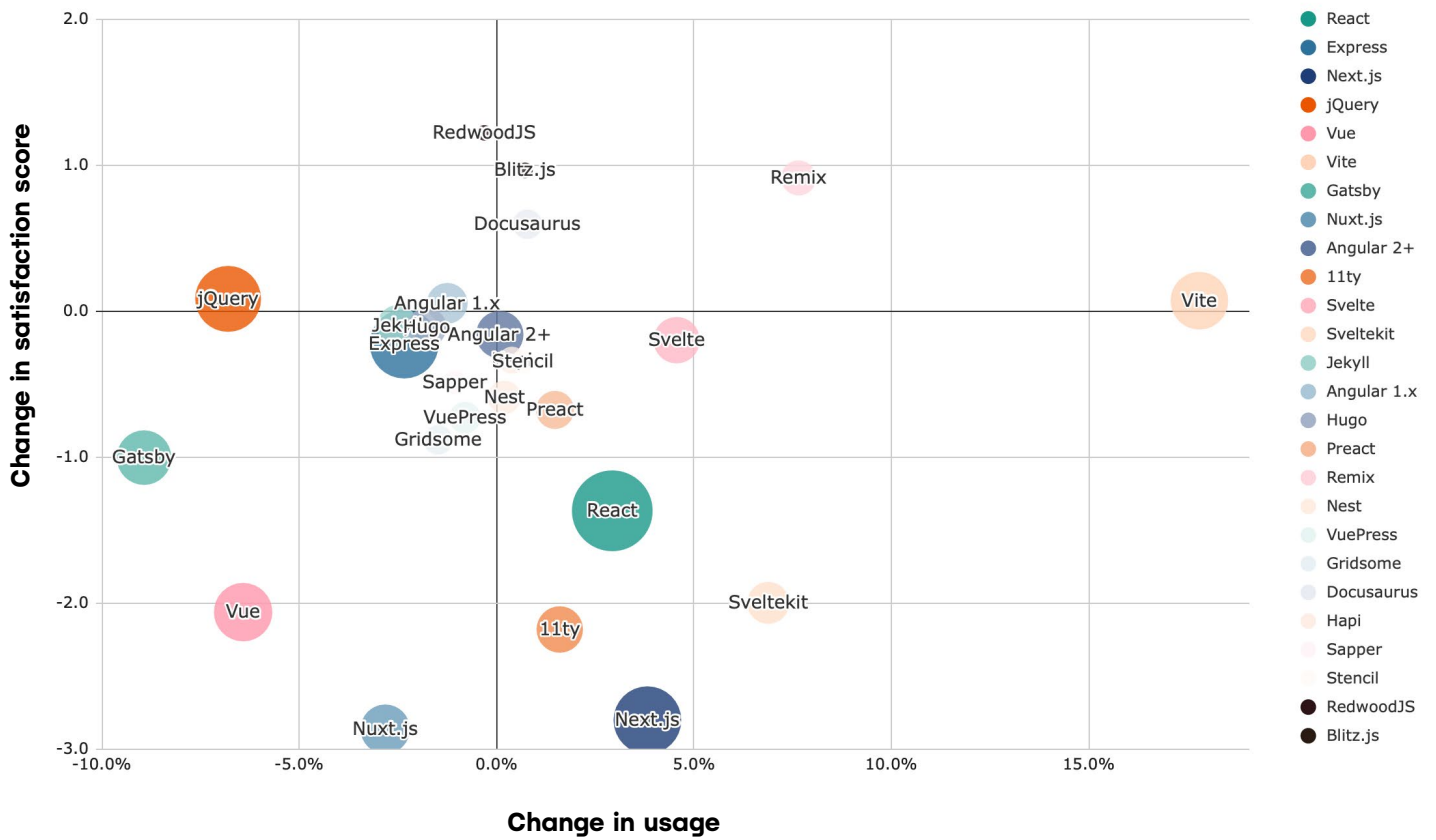


Want to Use Less
 Want to Use More

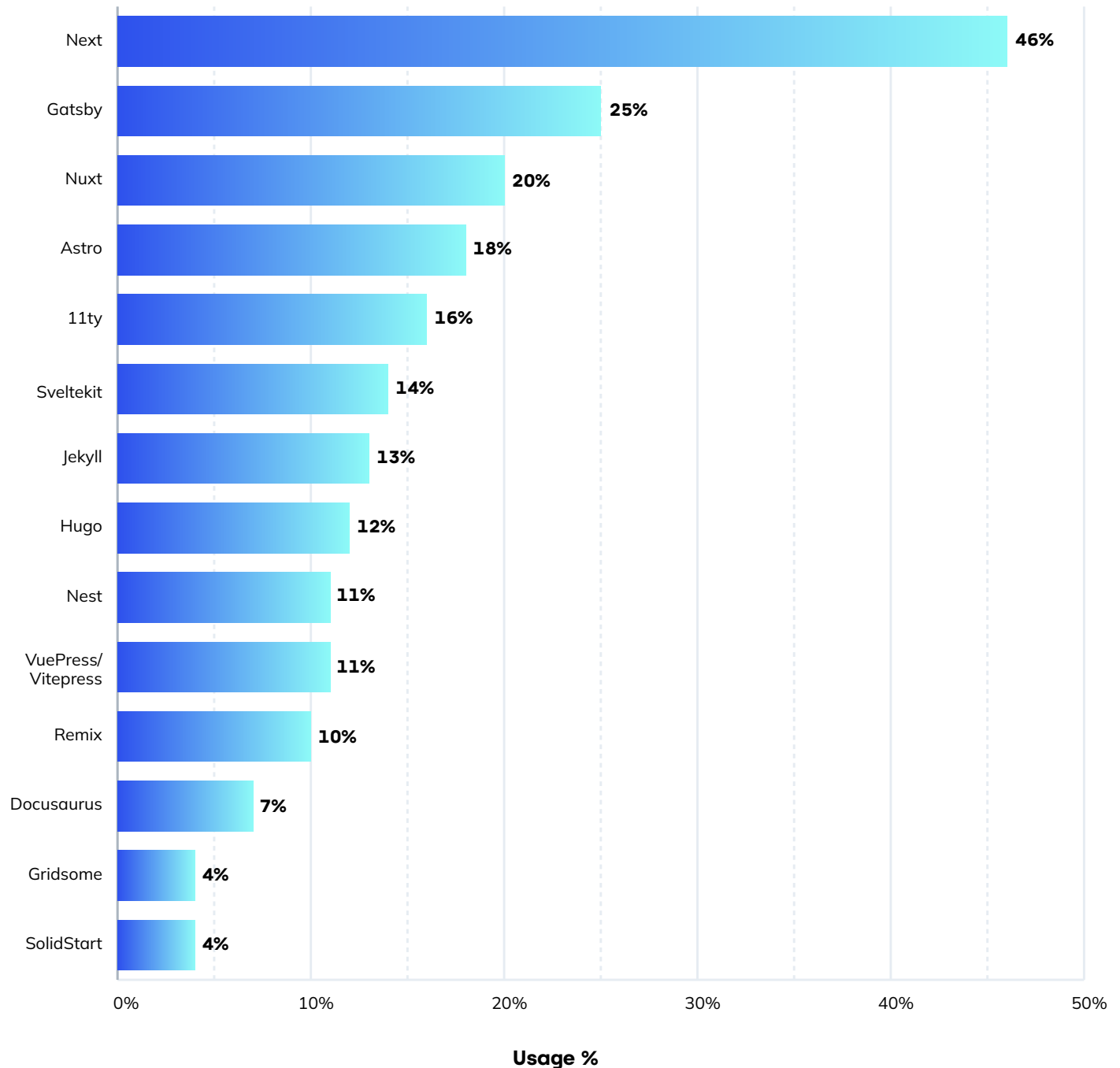
In the last 12 months, how much did you use the following front-end frameworks, and do you want to continue using them?



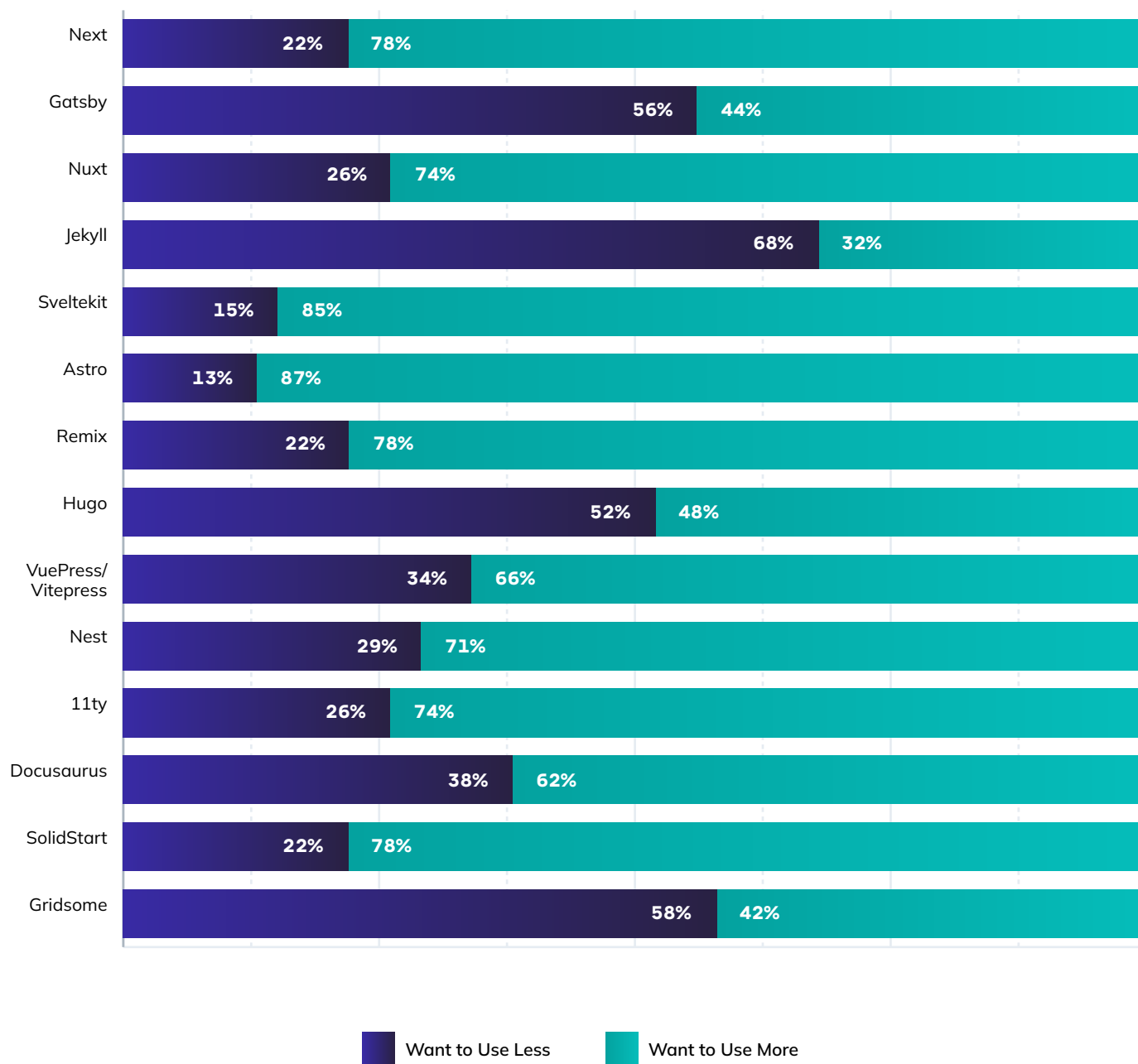
Front-end framework usage and satisfaction changes (2023 vs 2022)



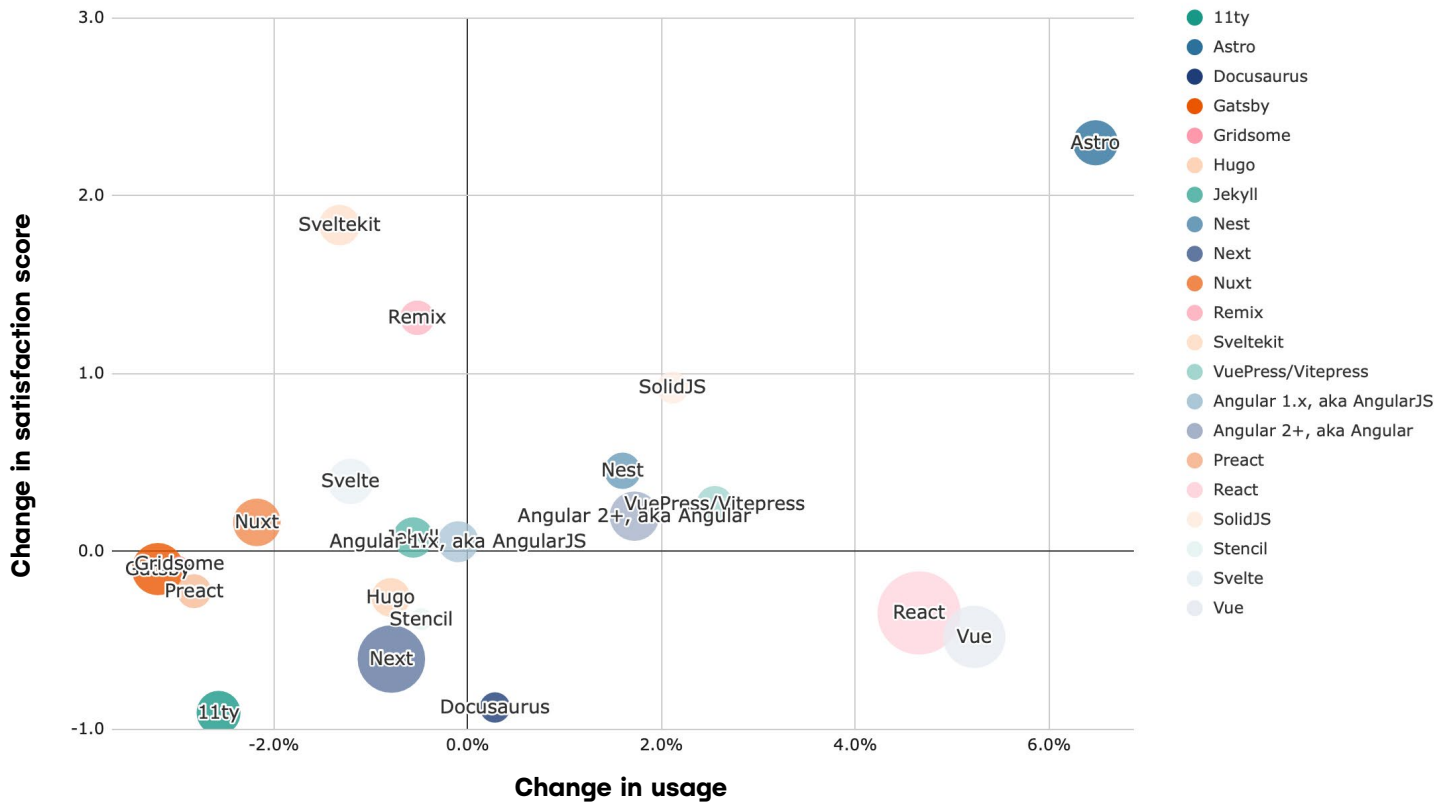
Thinking in the last 12 months, how much did you use the following rendering frameworks or site generators?



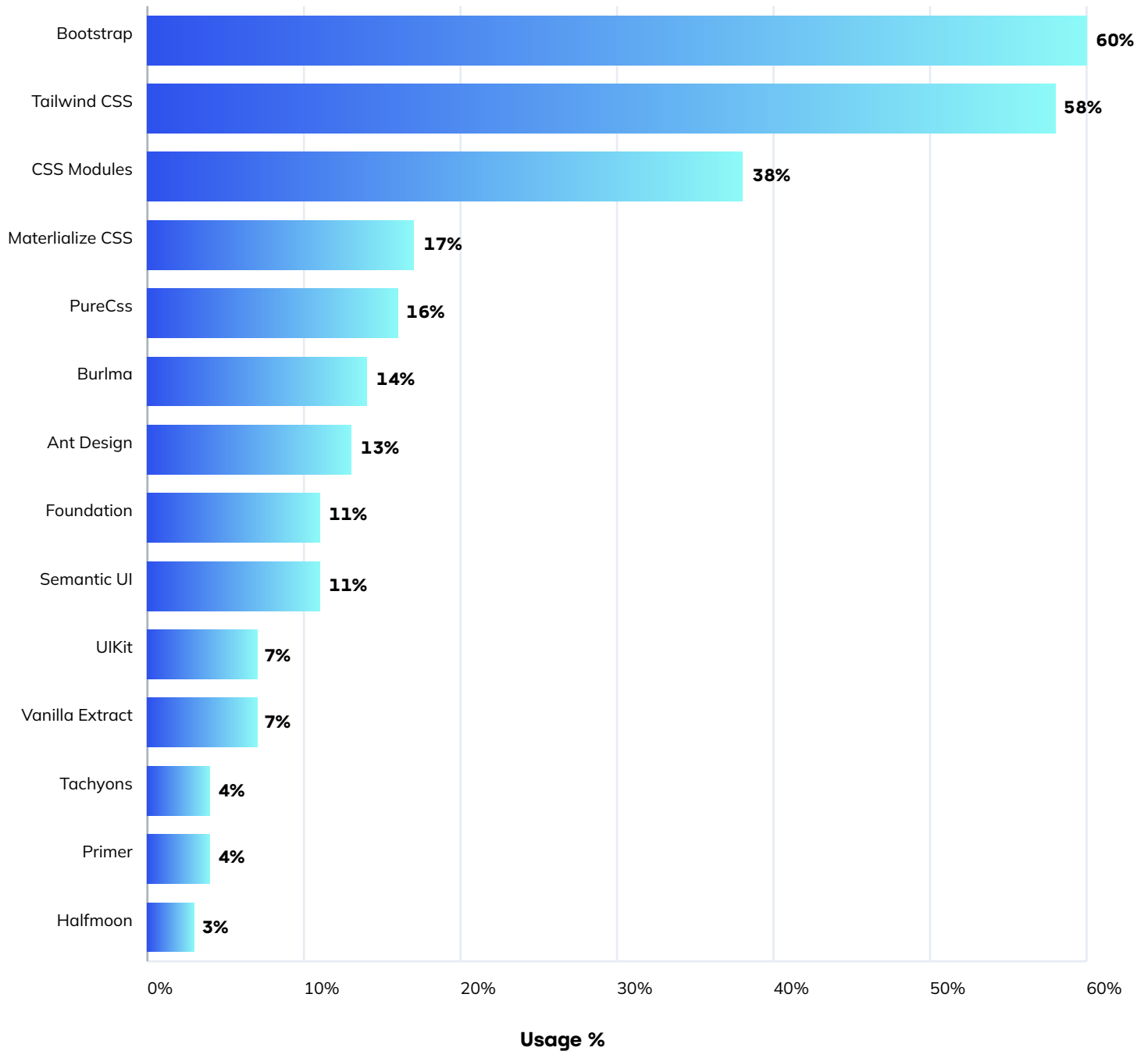
Thinking in the last 12 months, how much did you use the following rendering frameworks or site generators, and do you want to continue using them?



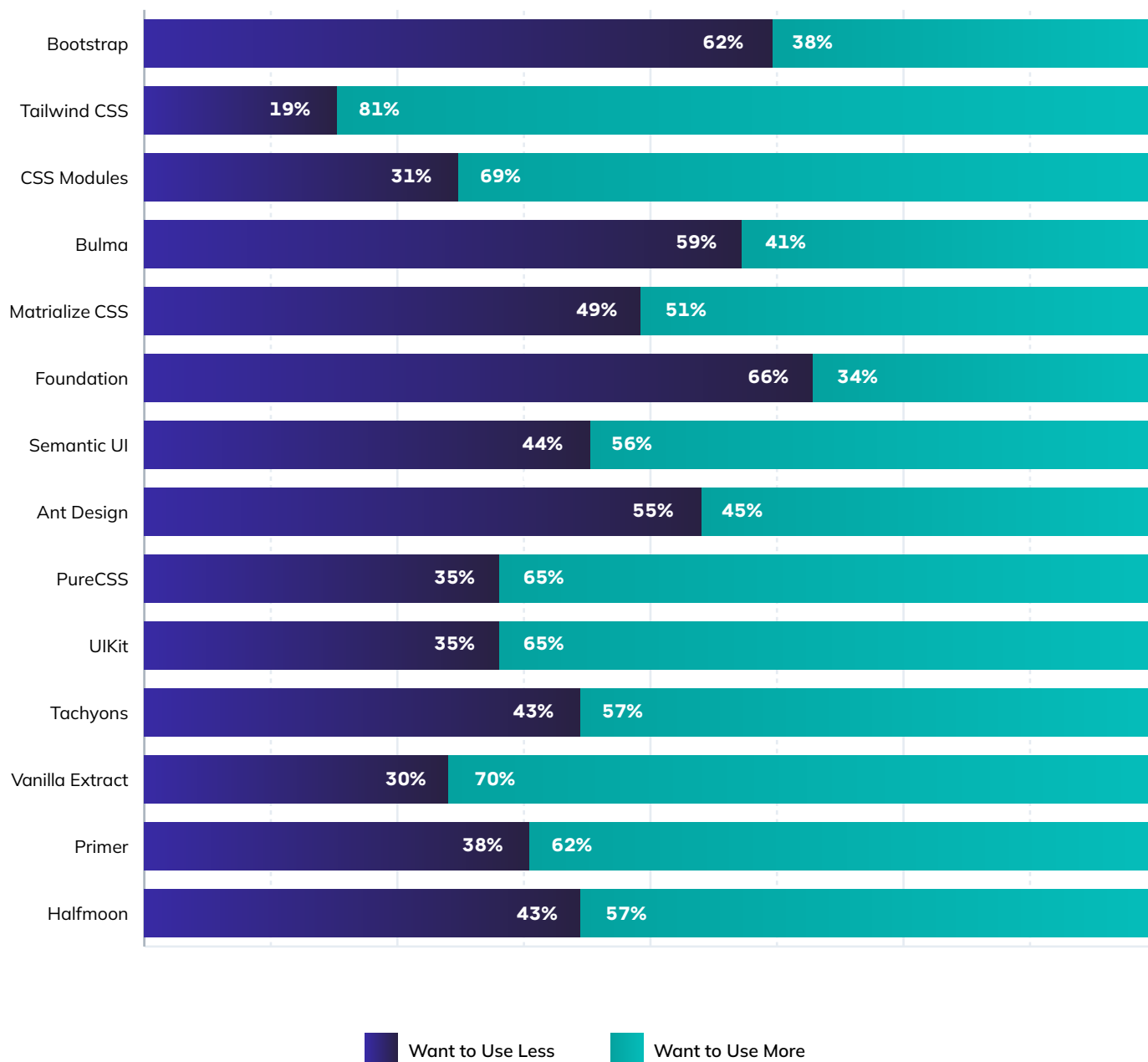
Rendering framework or site generator usage and satisfaction changes (2023 vs 2022)



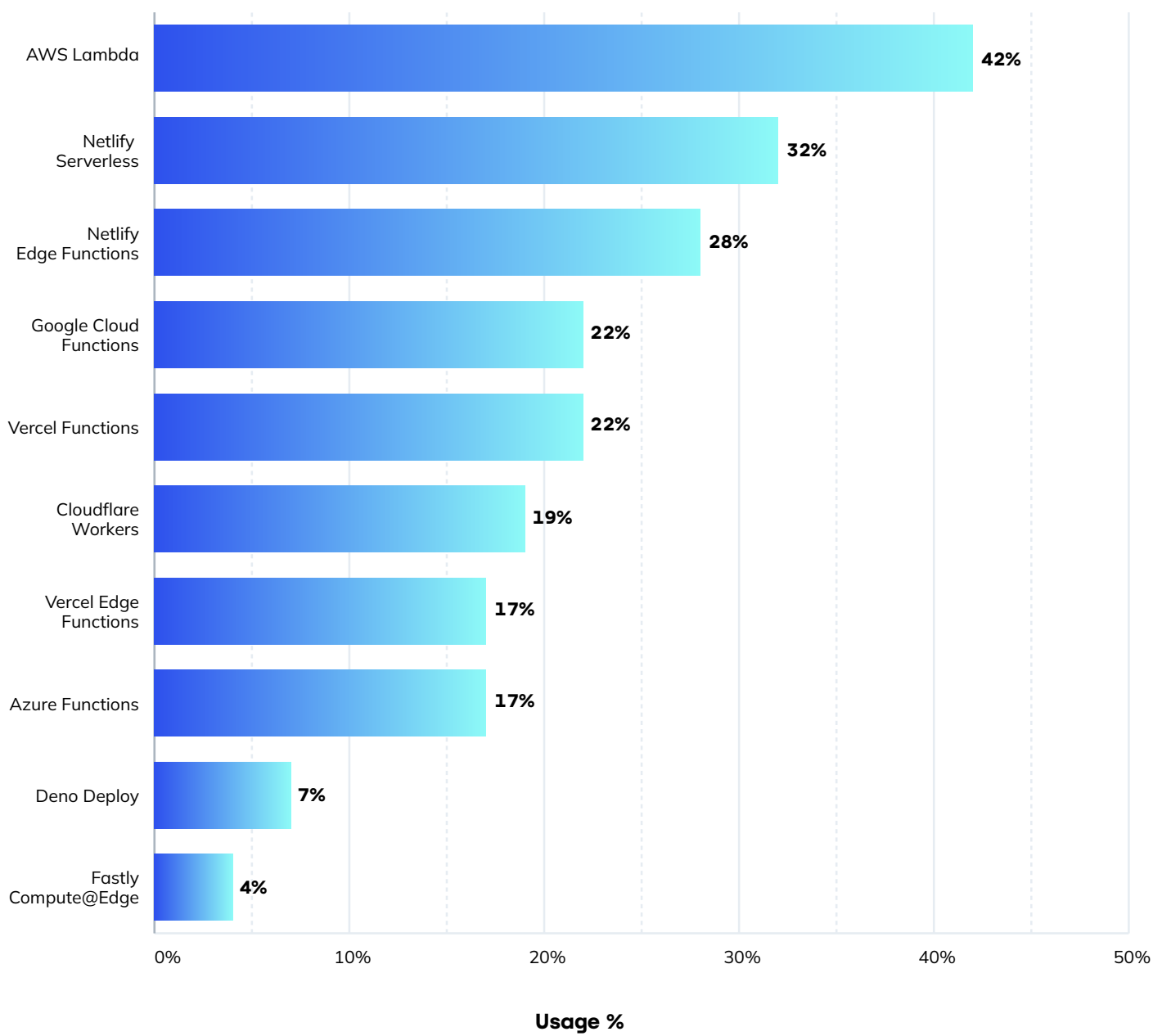
Thinking in the last 12 months, how much did you use the following CSS frameworks, and do you want to continue using them?



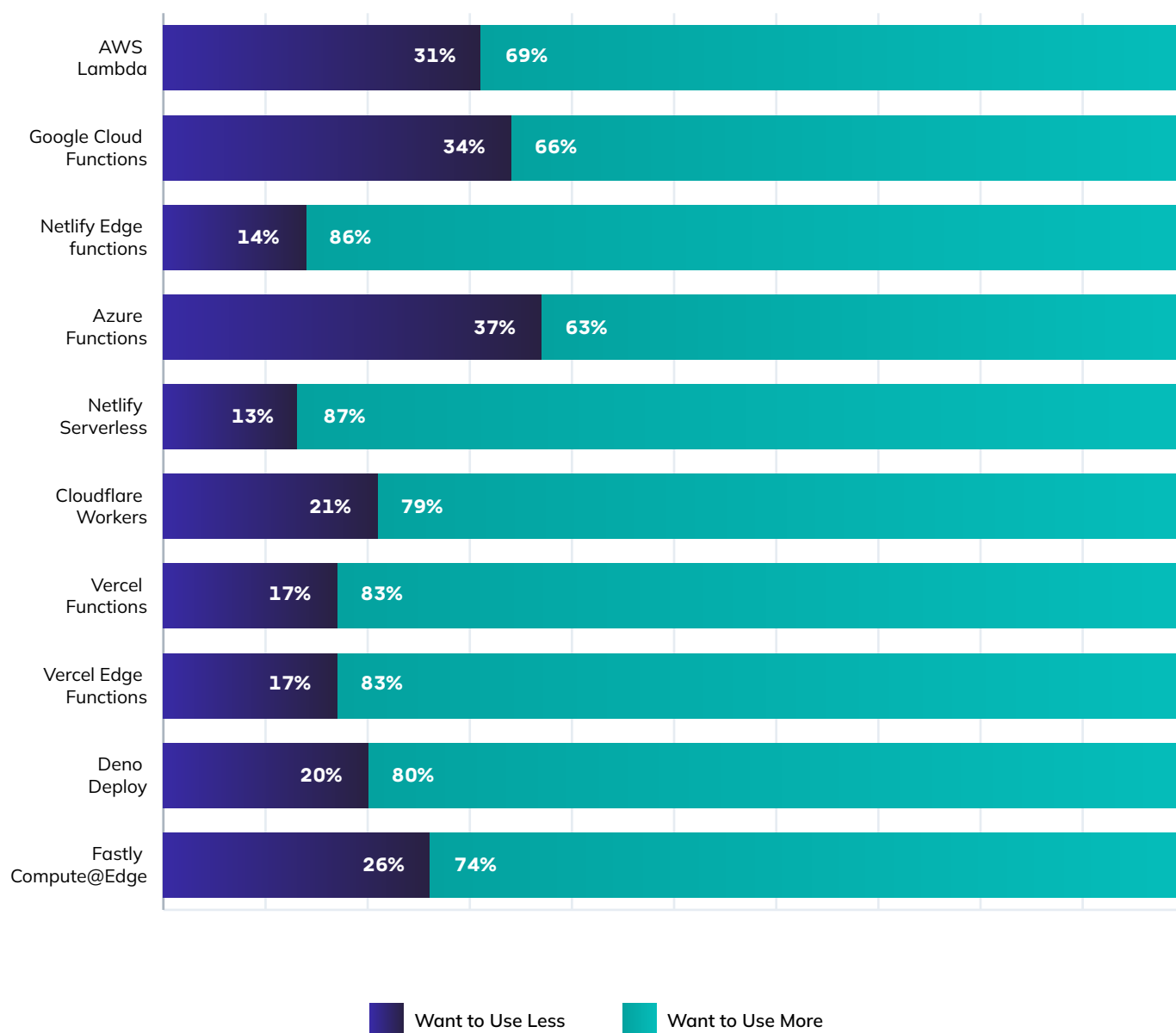
Thinking in the last 12 months, how much did you use the following CSS frameworks, and do you want to continue using them?



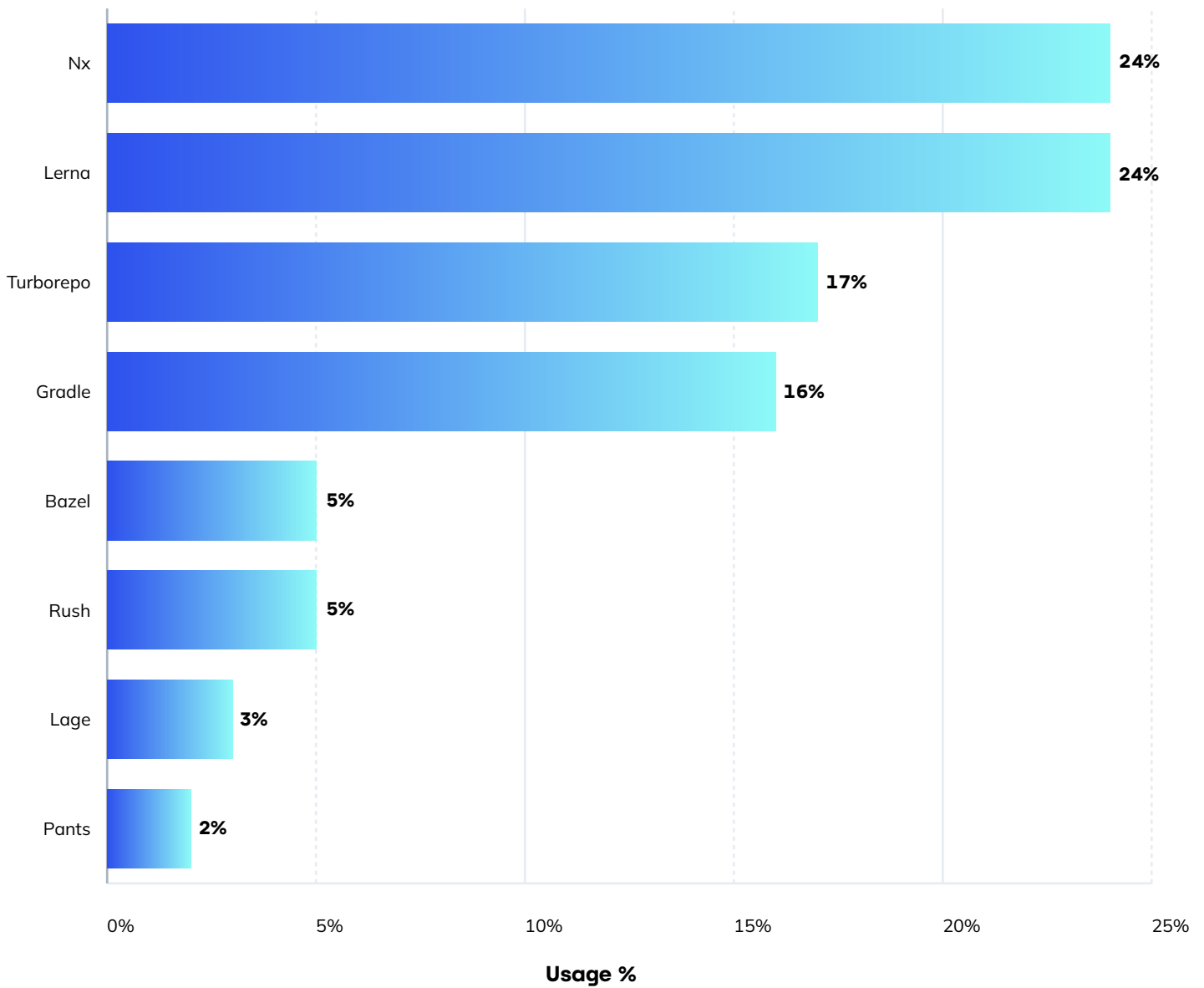
Thinking of sites you've worked on in the past 12 months, which serverless technology did you use?



Thinking in the last 12 months, which serverless technology have you used, and do you want to continue using them?



Thinking of sites you've worked on in the past 12 months, which monorepo tooling did you use?



Thinking of sites you've worked on in the past 12 months, which monorepo tooling have you use, and do you want to continue using them?

