O'REILLY®

Technology Trends for 2025

What O'Reilly Learning Platform Usage Tells Us About Where the Industry Is Headed

Mike Loukides

REPORT

Technology Trends for 2025
What O'Reilly Learning Platform Usage Tells Us About Where the Industry Is Headed

Mike Loukides

O'REILLY®

Technology Trends for 2025

by Mike Loukides

Editor: Mike Loukides	**Cover Designer:** Ellie Volckhausen
Copyeditor: Peyton Joyce	**Illustrator:** Ellie Volckhausen
Interior Designer: David Futato	

Table of Contents

Technology Trends for 2025

Welcome to our annual report on the usage of the O'Reilly learning platform. It's been an exciting year, dominated by a constant stream of breakthroughs and announcements in AI, and complicated by industry-wide layoffs. Generative AI gets better and better—but that trend may be at an end. Now the ball is in the application developers' court: Where, when, and how will AI be integrated into the applications we build and use every day? And if AI replaces the developers, who will be left to do the integration? Our data shows how our users are reacting to changes in the industry: Which skills do they need to brush up on? Which do they need to add? What do they need to know to do their day-to-day work? In short: Where have we been in the past year, and where are we going?

We aren't concerned about AI taking away software developers' jobs. Ever since the computer industry got started in the 1950s, software developers have built tools to help them write software. AI is just another tool, another link added to the end of that chain. Software developers are excited by tools like GitHub Copilot, Cursor, and other coding assistants that make them more productive.

That's only one of the stories we're following. Here are a few of the others:

- The next wave of AI development will be building agents: software that can plan and execute complex actions.
- There seems to be less interest in learning about programming languages, Rust being a significant exception. Is that because our users are willing to let AI "learn" the details of languages and libraries for them? That might be a career mistake.

- Security is finally being taken seriously. CEOs are tired of being in the news for the wrong reasons. AI tools are starting to take the load off of security specialists, helping them to get out of "firefighting" mode.

- "The cloud" has reached saturation, at least as a skill our users are studying. We don't see a surge in "repatriation," though there is a constant ebb and flow of data and applications to and from cloud providers.

- Professional development is very much of interest to our users. Specifically, they're focused on being better communicators and leading engineering teams.

All of these trends have been impacted, if not driven, by AI—and that impact will continue in the coming year.

Finally, some notes about methodology. Skip this paragraph if you want; we don't mind. This report is based on the use of O'Reilly's online learning platform from January 1, 2024, to September 30, 2024. Year-over-year comparisons are based on the same period in 2023. The data in each graph is based on O'Reilly's "units viewed" metric, which measures the actual use of each item on the platform. It accounts for different usage behavior for different media: text, courses, and quizzes. In each graph, the data is scaled so that the item with the greatest units viewed is 1. That means items within a graph are comparable to each other, but you can't compare an item in one graph to an item in another. And all percentages are reported with two significant digits.

Skills

When we look at how our customers use the O'Reilly learning platform, we always think in terms of skills. What skills are they trying to gain? And how are they trying to improve their knowledge? This year, one thread that we see across all of our platform is the importance of artificial intelligence. It's all about upskilling in the age of AI.

Artificial Intelligence

It will surprise absolutely nobody that AI was the most active category in the past year. For the past two years, large models have dominated the news. That trend started with ChatGPT and its descendants, most recently GPT 4o1. But unlike 2022, when ChatGPT was the only show anyone cared about, we now have many contenders. Claude has emerged as a favorite among programmers. After a shaky start, Google's Gemini models have become solid performers. Llama has established itself as one of the top models and as the matriarch of a rich ecosystem of open[1] models. Many of the open models can deliver acceptable performance when running on laptops and phones; some are even targeted at embedded devices.

So what does our data show? First, interest in almost all of the top skills is up: From 2023 to 2024, Machine Learning grew 9.2%; Artificial Intelligence grew 190%; Natural Language Processing grew 39%; Generative AI grew 289%; AI Principles grew 386%; and Prompt Engineering grew 456%. Among the top topics, the most significant decline was for GPT itself, which dropped by 13%—not a huge decline but certainly a significant one. Searches for GPT peaked in March 2023 and have been trending downward ever since, so our search data matches our usage data.

We're used to seeing interest move from a more general high-level topic to specific skills as an industry sector matures, so this trend away from GPT in favor of more abstract, high-level topics is counterintuitive. But in context, it's fairly clear what happened. For all practical purposes, GPT was the only game in town back in 2023. The situation is different now: There's lots of competition. These other models don't yet show up significantly in search or usage data, but the users of our platform have figured out what's important: not learning about GPT or Claude or Gemini or Mistral but getting the background you need to make sense of any model. Discovering a workflow that fits your needs is important, and as Simon Willison points out, your ideal workflow may actually involve using several models. Recent models are all good, but they aren't all good in the same way.

---

1 The definition of "open" and "open source" for AI is still controversial. Some open models don't include access to weights, and many don't include access to training data.

AI has had a great year, but will it continue to show gains in 2025? Or will it drop back, much as ChatGPT and GPT did? That depends on many factors. Gartner has generative AI slipping into the "trough of disillusionment"—and whatever you think of the technology's promise, remember that the disillusionment is a sociological phenomenon, not a technical one, and that it happens because new technologies are overhyped. Regardless of generative AI's long-term promise, we expect some disillusionment to set in, especially among those who haven't properly understood the technology or its capabilities.

Prompt Engineering, which gained 456% from 2023 to 2024, stands out. A 456% gain isn't as surprising as it seems; after all, people only started talking about prompt engineering in 2023. Although "prompt engineering" was bandied about as a buzzword, it didn't become a skill that employers were looking for until late in 2023, if that. That may be an early warning signal for AI disillusionment. Searches for "prompt engineering" grew sharply in 2023 but appeared to decline slightly in 2024. Is that noise or signal? If disillusionment in Prompt Engineering sets in, we'll also see declines in higher-level topics like Machine Learning and Artificial Intelligence.

There's a different take on the future of prompt engineering. There have been a number of arguments that the need for prompt engineering is temporary. As generative AI improves, this line of reasoning contends, we will no longer need to write complex prompts that specify exactly what we want the AI to do and how to do it. Prompts will be less sensitive to exactly how they're worded; changing a word or two will no longer give a completely different result. We'll no longer have to say "explain it to me as if I were five years old" or provide several examples of how to solve a problem step-by-step.

Some recent developments point in that direction. Several of the more advanced models have made the "explain it to me" prompts superfluous. OpenAI's GPT 4o1 has been trained in a way that maximizes its problem-solving abilities, not just its ability to string together coherent words. At its best, it eliminates the need to write prompts that demonstrate how to solve the problem (a technique called few-shot prompting). At worst, it "decides" on an inappropriate process, and it's difficult to convince it to solve the problem a different way. Anthropic's Claude has a new (beta) computer use feature that lets the model use browsers, shells, and other programs: It can click on links and buttons, select text, and

do much more. (Google and OpenAI are reportedly working on similar features.) Enabling a model to use the computer in much the same way as a human appears to give it the ability to solve multistep problems on its own, with minimal description. It's a big step toward a future full of intelligent agents: linked AI systems that cooperate to solve complex problems. However, Anthropic's documentation is full of warnings about serious security vulnerabilities that remain to be solved. We're thrilled that Anthropic has been forthright about these weaknesses. But still, while computer use may be a peek at the future, it's not ready for prime time.

AI will almost certainly slide into a trough of disillusionment; as I've said, the trough has more to do with sociology than with technology. But OpenAI and Anthropic are demonstrating important paths forward. Will these experiments bear fruit in the next year? We'll see.
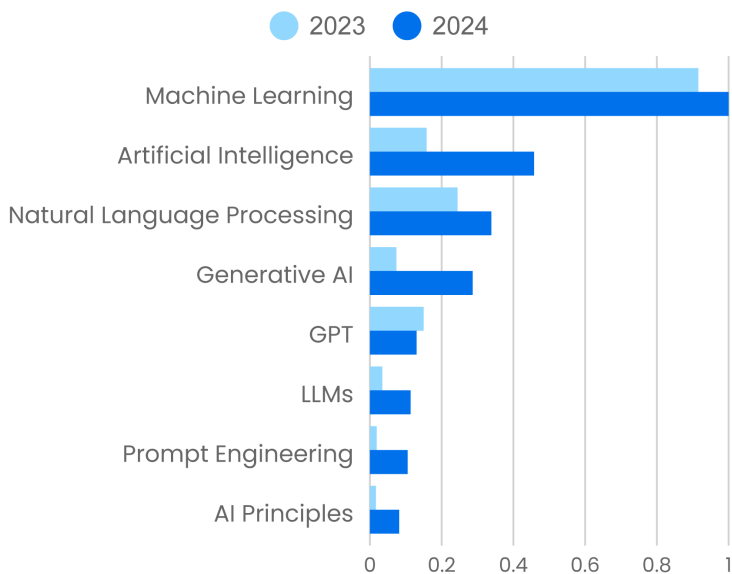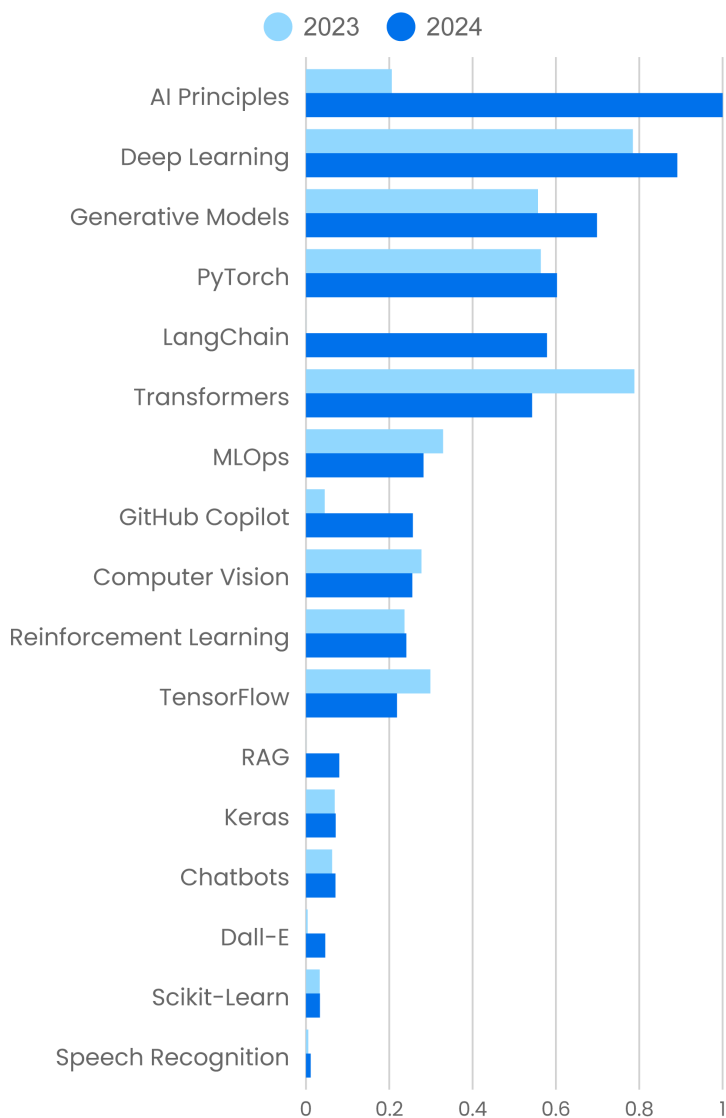

Figure 1. Artificial intelligence

Many skills associated with AI also showed solid gains. Use of content about Deep Learning is up 14%, Generative Models is up 26%, and GitHub Copilot is up 471%. Use of content about the major AI libraries was up slightly: PyTorch gained 6.9%, Keras increased 3.3%, and Scikit-Learn gained 1.7%. Usage of TensorFlow content

declined 28%; its continued decline indicates that PyTorch has won the hearts and minds of AI developers.

These gains—particularly Copilot's—are impressive, but a more important story concerns two skills that came out of nowhere: Usage of content about LangChain is on a par with PyTorch, and RAG is on a par with Keras. Neither of these skills were in last year's report; in 2023, content usage for LangChain and RAG was minimal, largely because little content existed. They've caught on because both LangChain and RAG are tools for building better applications on top of AI models. GPT, Claude, Gemini, and Llama aren't the end of the road. RAG lets you build applications that send private data to a model as part of the prompt, enabling the model to build answers from data that wasn't in its training set. This process has several important consequences: It minimizes the probability of error or "hallucination"; it makes it possible to attribute answers to the sources from which they came; and it often makes it possible to use a much smaller and more economical model.

LangChain is the first of many frameworks for building AI agents. (OpenAI has Swarm; Google has an Agent Builder that's part of Vertex; Salesforce and other vendors also have offerings.) Agents are software that can plan and execute multistage actions, many of which are delegated to other AI models. Claude's computer use API is another facet of this trend, along with whatever products OpenAI and Google may be building. Saying that usage has increased 26 million percent isn't to the point—but realizing that LangChain has grown from near zero to a platform on a par with PyTorch is very much so. Agentic applications are certainly the next big trend within AI.

Figure 2. Skills needed for AI

Data

Artificial intelligence relies heavily on what we used to call (and perhaps still call) data science. Building AI models requires data at unprecedented scale. Building applications with RAG requires a portfolio of data (company financials, customer data, data purchased from other sources) that can be used to build queries, and data scientists know how to work with data at scale.

Therefore, it's not surprising that Data Engineering skills showed a solid 29% increase from 2023 to 2024. SQL, the common language of all database work, is up 3.2%; Power BI was up 3.0%, along with the more general (and much smaller) topic Business Intelligence (up 5.0%). PostgreSQL is close to edging ahead of MySQL, with a 3.6% gain. Interest in Data Lake architectures rose 59%, while the much older Data Warehouse held steady, with a 0.3% decline. (In our skill taxonomy, Data Lake includes Data Lakehouse, a data storage architecture that combines features of data lakes and data warehouses.) Finally, ETL grew 102%. With the exception of ETL, the gains are smaller than the increases we saw for AI skills, but that makes sense: AI is an exciting new area, and data is a mature, stable category. The number of people who need specialized skills like ETL is relatively small but obviously growing as data storage becomes even more important with AI.

It's worth understanding the connection between data engineering, data lakes, and data lakehouses. Data engineers build the infrastructure to collect, store, and analyze data. The data needed for an AI application almost always takes many forms: free-form text, images, audio, structured data (for example, financial statements), etc. Data often arrives in streams, asynchronously and more or less constantly. This is a good match for a data lake, which stores data regardless of structure for use later. Because data receives only minimal processing when it arrives, it can be stored in near real time; it's cleaned and formatted in application-specific ways when it's needed. Once data has been stored in a data lake, it can be used for traditional business analytics, stored in a vector or graph database for RAG, or put to almost any other use. A data lakehouse combines both structured and unstructured data in a single platform.
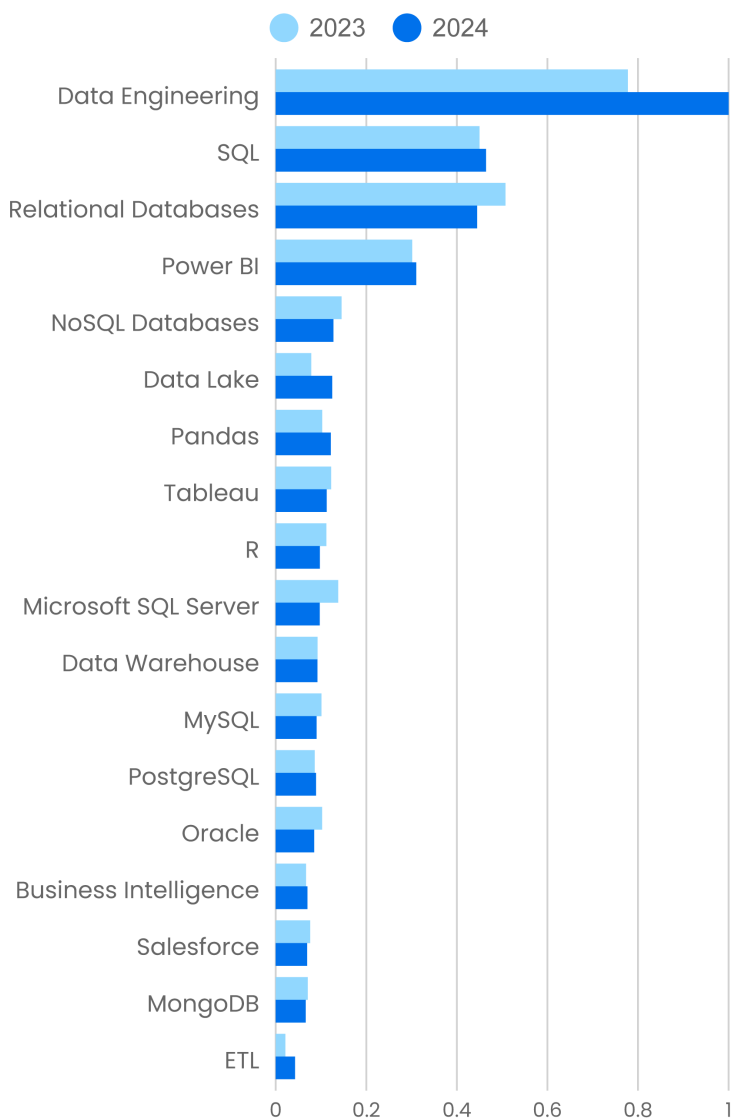
Figure 3. Data analysis (including databases)

Software Development

What do software developers do all day? They write software. Programming is an important part of the job, but it's not the whole thing; best estimates are that programmers spend roughly 20% of their time writing code. The rest of their time is spent understanding the problems they're being asked to solve, designing appropriate solutions, documenting their work, updating management on the status of their projects, and much more.

Software architecture, which focuses on understanding a customer's requirements and designing systems to meet those requirements, is an important part of the overall software development picture. It's a skill to which many of our software developers and programmers aspire.

Architecture

This year's data shows that software architecture continues to be one of the most desirable skills in the industries we serve. Usage of material about Software Architecture rose 5.5% from 2023 to 2024, a small but significant increase. But it's more important to ask why it increased. A position in software architecture may be perceived as more secure in a time of layoffs, and it's often perceived as another step forward in a career that moves from junior programmer to senior to lead. In addition, the rise of AI presents many architectural challenges: Do we try to build our own model? (The answer is usually "no.") Should we use an AI service provider like OpenAI, Anthropic, Microsoft, or Google, or should we fine-tune and host our own model on our own infrastructure? How do we build applications that are safe (and how do we define "safe")? How do we evaluate performance? These questions all have a bearing on software architecture. Furthermore, AI might provide tools to help software architects, but so far, these tools can do little for the substance of the job: understanding customers' needs and helping them define what they want to build. With AI in the picture, we're all building new kinds of applications—and those applications require architects to help design them.

In this context, it's no surprise that Enterprise Architecture is up 17% and Distributed Systems is up 35%. Enterprise architecture is a staple: As Willie Sutton said about banks, "That's where the money is." It's a good bet that many enterprises are trying to integrate

AI into their systems or update legacy systems that are no longer scalable or maintainable. We can (and do) make the same argument about distributed systems. Modern enterprises work on a scale that was unimaginable a few decades ago. Scale isn't just for companies like Amazon and Google. To survive, even small businesses need to develop an online presence—and that means building systems in the cloud that can handle surges in demand gracefully. It means building systems that can withstand outages. Distributed systems aren't just massive deployments with hundreds of thousands of nodes. Your business may only require a dozen nodes, but regardless of the scale, it still faces the architectural challenges that come with distributed systems.

Some of the more significant ideas from the past decade seem to be falling out of favor. Microservices declined 24%, though content use is still substantial. Domain-Driven Design, which is an excellent skill for designing with microservices, is down 22%. Serverless is down 5%; this particular architectural style was widely hyped and seemed like a good match for microservices but never really caught on, at least based on our platform's data.

What's happening? Microservice architectures are difficult to design and implement, and they aren't always appropriate—from the start, the best advice has been to begin by building a monolith, then break the monolith into microservices when it becomes unwieldy. By the time you reach that stage, you'll have a better feel for what microservices need to be broken out from the monolith. That's good advice, but the hype got ahead of it. Many organizations that would never need the complexity of microservices were trying to implement them with underskilled staff. As an architectural style, microservices won't disappear, but they're no longer getting the attention they once were. And new ideas, like modular monoliths, may catch on in the coming years; modularity is a virtue regardless of scale or complexity.
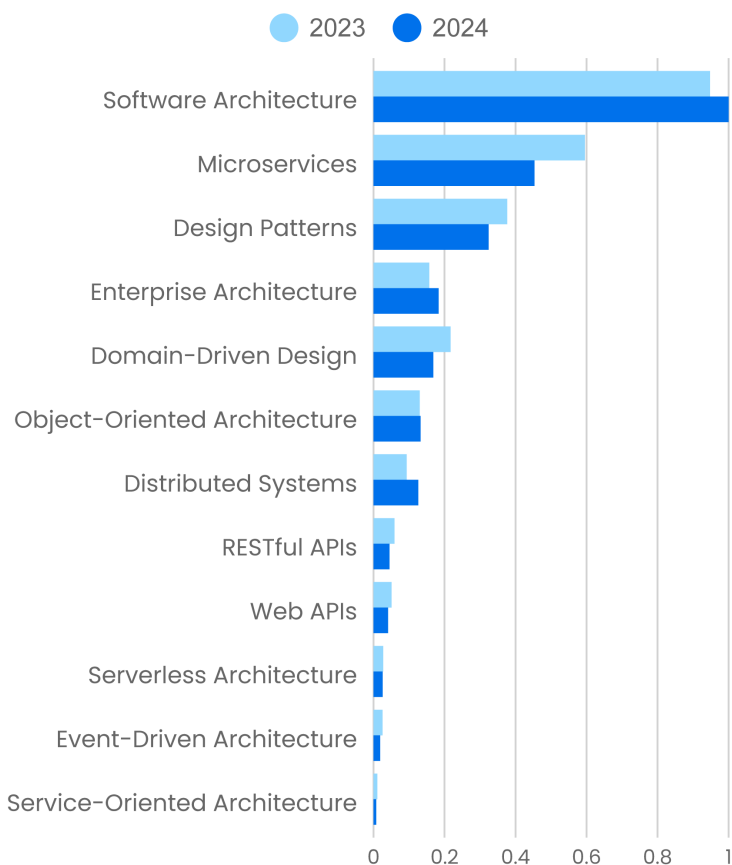
Figure 4. Software architecture and design

Programming languages

Last year's report showed that our users were consuming less content about programming languages. This year's data continues that trend. We see a small drop for Python (5.3%) and a more significant drop for Java (13%). And even C++, which showed healthy growth from 2022 to 2023, is down 9% in 2024.

On the other hand, C is up (1.3%), and so is C# (2.1%). Rust is up 9.6%. The small increases in C and C# may just be noise. C is well-entrenched and isn't going anywhere fast. Neither is C++, despite its drop. Rust's increase continues a growth trend that stretches back several years; that's an important signal. Rust is clearly winning over

developers, at least for new projects. Now that the US government is placing a priority on memory safety, Rust's emphasis on memory safety serves it well. Rust isn't the first programming language to claim memory safety, nor will it be the last. (There are projects to add memory safety to C++, for example.) But right now, it's the best positioned.

Aside from Rust, though, we need to ask what's happening with programming skills. A few forces are applying downward pressure. Industry-wide layoffs may be playing a role. We've downplayed the effect of layoffs in the past, but we may have to admit that we were wrong: This year, they may be taking a bite out of skills development.

Could generative AI have had an effect on the development of programming language skills? It's possible; shortly after GPT-3 was released, Simon Willison reported that he was learning Rust with the help of ChatGPT and Copilot, and more recently that he's used Claude to write Rust code that he has in production, even though he doesn't consider himself a skilled Rust developer.

It would be foolish to deny that generative AI will help programmers to become more productive. And it would be foolish to deny that AI will change how and what we learn. But we have to think carefully about what "learning" means, and why we learn in the first place. Programmers won't have to remember all the little details of programming languages—but that's never been the important part of programming, nor has rote memorization been an important part of learning. Students will never have to remember a half dozen sorting algorithms, but computer science classes don't teach sorting algorithms because committing algorithms to memory is important. Every programming language has a sort() function somewhere in its libraries. No, sorting is taught because it's a problem that everyone can understand and that can be solved in several different ways—and each solution has different properties (performance, memory use, etc.). The point is learning how to solve problems and understanding the properties of those solutions. As Claire Vo said in her episode of *Generative AI in the Real World*, we'll always need engineers who think like engineers—and that's what learning how to solve problems means. Whether lines end in a semicolon or a colon or whether you use curly braces, end statements, or tabs to delimit blocks of code is immaterial.
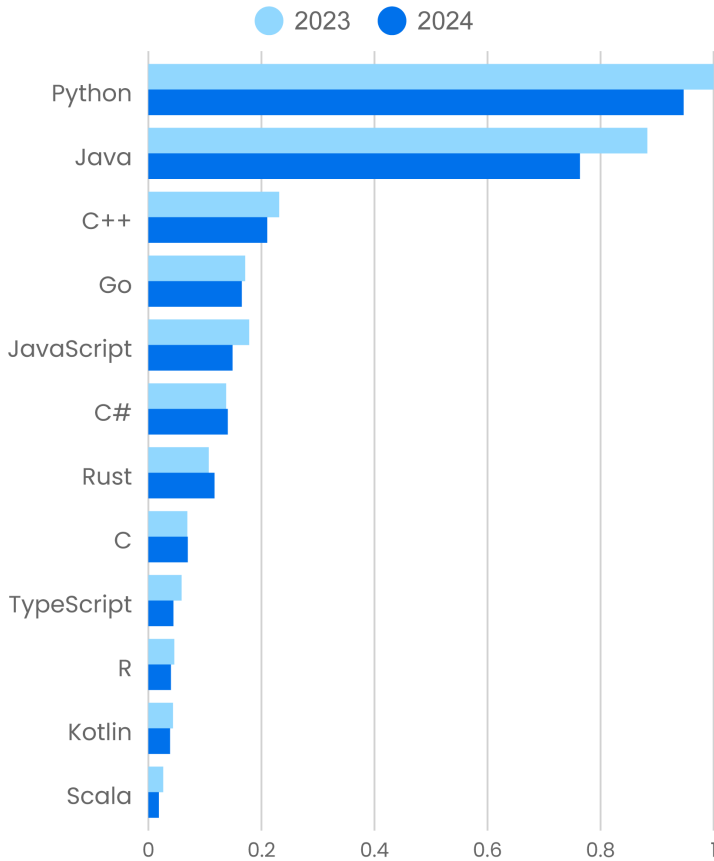
Figure 5. Programming languages

The perception that generative AI minimizes the need to learn programming languages may limit the use of language-oriented content on our platform. Does that benefit the learners? If someone is using AI to avoid learning the hard concepts—like solving a problem by dividing it into smaller pieces (like quicksort)—they are shortchanging themselves. Shortcuts rarely pay off in the long term; coding assistants may help you to write some useful code, but those who use them merely as shortcuts rather than as learning tools are missing the point. Unfortunately, the history of teaching—going back centuries if not millennia—has stressed memorization. It's time for both learners and teachers to grow beyond that.

Learning is changing as a result of AI. The way we teach, and the way our users want to be taught, is changing. Building the right kind of experiences to facilitate learning in an AI-enabled environment is an ongoing project for our learning platform. In the future, will our users learn to program by completing AI-generated tutorials that are customized in real time to their needs and abilities? That's where we're headed.

Web programming

Use of content about web programming skills is down, with few exceptions. A number of factors might be contributing to this. First, I can't think of any significant new web frameworks in the past year; the field is still dominated by React (down 18%) and Angular (down 10%). There is some life near the bottom of the chart. The Svelte framework had significant growth (24%); so did Next.js (8.7%). But while these frameworks have their adherents, they're far from dominant.

PHP (down 19%) still claims to have built the lion's share of the web, but it's not what developers reach for when they want to build something new, particularly if that "new" is a complex web application. The PHP world has been rocked by a bitter fight between the CEOs of Automattic (the developers of WordPress, by far the most important PHP framework) and WP Engine (a WordPress hosting platform). That fight started too late to affect this year's results significantly, but it might weigh heavily next year.

A more significant development has been the movement away from complex platforms and back toward the simplicity of the earlier web. Alex Russell's "Reckoning" posts summarize many of the problems. Our networks and our computers are much, much faster than they were 20 or 25 years ago, but web performance hasn't improved noticeably. If anything, it's gotten worse. We still wait for applications to load. Applications are hard to develop and have gotten harder over the years. There are several new frameworks that may (or may not) be lighter-weight, such as HTMX, Ludic, Glitch, and Cobalt. None of them have yet made a dent in our data, in part because none have built enough of a following for publishers and trainers to develop content—and you can't have any units viewed if there isn't anything to view. However, if you want an experience that isn't dominated by heavyweight frameworks, doesn't require you to

become a JavaScript expert, and puts the fun back into building the web, this is where to look.
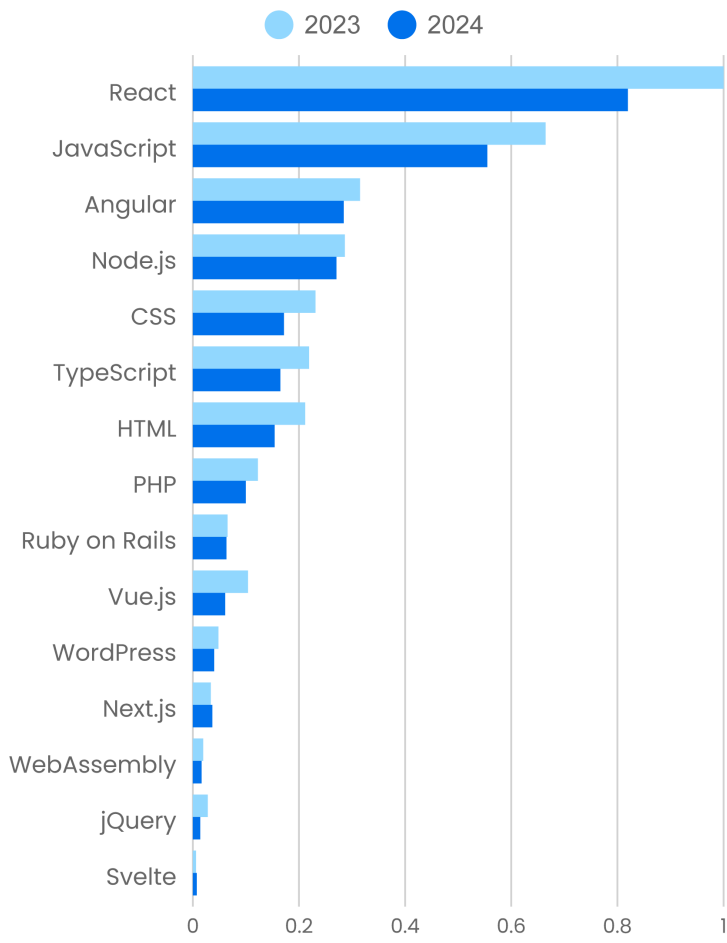
Figure 6. Web development

Web dev is a discipline that has been ill-served by shortcuts to learning. We hear too often about boot camp graduates who know a few React tricks but don't understand the difference between React and JavaScript (or even know that JavaScript exists, let alone other programming languages). These programmers are very likely to lose their jobs to AI, which can already reproduce all the basic React techniques they've learned. Learning providers need to think about how AI is changing the workplace and how their students can

partner with AI to build something beyond what AI can build on its own. Part of the solution is certainly a return to basics, ensuring that junior developers understand the tools with which they're working.

IT Operations

Operations is another area where the trends are mostly downward. It may be small consolation, but the drops for several of the most important topics are relatively small: Linux is down 1.6%, Terraform is down 4.0%, and Infrastructure as Code is down 7.3%. As a skill, Terraform seems little hurt by the fork of Terraform that created the open source OpenTofu project, perhaps because the OpenTofu developers have been careful to maintain compatibility with Terraform. How this split plays out in the future is an open question. It's worth noting the precipitous drop in Terraform certification (down 43%); that may be a more important signal than Terraform itself.

Kubernetes is down 20%. Despite that drop, which is sharper than last year's 6.9% decrease, content teaching Kubernetes skills remains the second most widely used group in this category, and Kubernetes certification is up 6.3%. Last year, we said that Kubernetes needed to be simpler. It isn't. There are no viable alternatives to Kubernetes yet, but there are different ways to deploy it. Kubernetes as a service managed by a cloud provider is certainly catching on, putting the burden of understanding every detail of Kubernetes's operation on the shoulders of the provider. We also pointed to the rise of developer platforms; this year, the buzzword is "platform engineering" (Camille Fournier and Ian Nowland's book is excellent), but as far as Kubernetes is concerned, it's the same thing. Platform engineers can abstract knowledge of Kubernetes into a platform, minimizing software developers' cognitive overhead. The result is that the number of people who need to know about Kubernetes is smaller.

Both DevOps (down 23%) and SRE (down 15%) dropped. There's certainly some frustration with DevOps: Has it paid off? We ask a different question: Has it ever been tried? One problem with DevOps (which it shares with Agile) is that many companies "adopted" it in name but not in essence. They renamed a few positions, hired a few DevOps engineers, maybe created a DevOps group, never realizing that DevOps wasn't about new job titles or new specialties; it was about reducing the friction between software development teams and operations teams. When you look at it this way, creating new groups and hiring new specialists can only be

counterproductive. And the result is predictable: You don't have to look far to find blogs and whitepapers claiming that DevOps doesn't work. There's also frustration with ideas like "shift left" and DevSecOps, which envision taking security into account from the start of the development process. Security is a different discussion, but it's unclear how you build secure systems without taking it into account from the start. We've spent several decades building software and trying to fold security in at the last minute—we know how well that works.
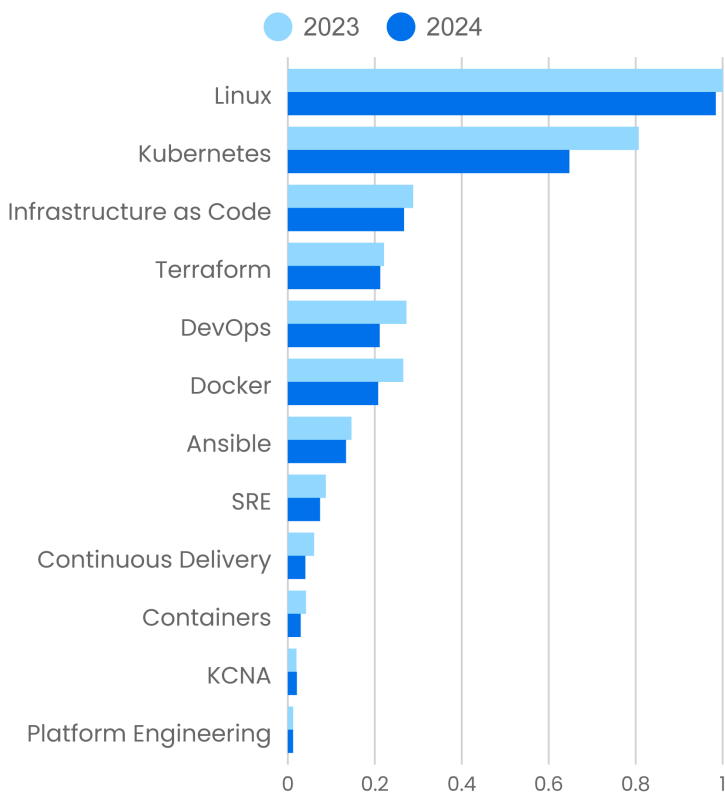

Figure 7. Infrastructure and operations

In any case, the industry has moved on. Platform engineering is, in many ways, a natural outgrowth of both DevOps and SRE. As I've argued, the course of operations has been to increase the ratio of computers to operators. Is platform engineering the next step, allowing software developers to build systems that can handle their

own deployment and routine operations without the help of operations staff?

IT certifications

General IT certifications, apart from security, trended downward. Use of content to prepare for the CompTIA A+ exam, an entry-level IT certification, was down 15%; CompTIA Network+ was down 7.9%. CompTIA's Linux+ exam held its own, with a decline of 0.3%. On our platform, we've seen that Linux resources are in high demand. The slight decline for Linux-related content (1.6%) fits with the very small decrease in Linux+ certification.

For many years, Cisco's certifications have been the gold standard for IT. Cisco Certified Network Associate (CCNA), a fairly general entry-level IT certification, showed the greatest usage and the smallest decline (2.2%). Usage of content to prepare for the Cisco Certified Network Practitioner (CCNP) exams, a cluster of related certifications on topics like enterprise networking, data centers, and security, dropped 17%. The Cisco Certified Internet Engineer (CCIE) exams showed the greatest decline (36%). CCIE has long been recognized as the most comprehensive and in-depth IT certification. We're not surprised that the total usage of this content is relatively small. CCIE represents the climax of a career, not the start. The number of people who attain it is relatively small, and those who do often include their CCIE number with their credentials. But the drop is surprising. It's certainly true that IT is less focused on heavy-duty routing and switching for on-prem data centers (or even smaller machine rooms) than it was a few years ago. That work has largely been offloaded to cloud providers. While routers and switches haven't disappeared, IT doesn't need to support as wide a range of resources: They need to support office WiFi, some databases that need to remain on-premises, and maybe a few servers for office-related tasks. They're very concerned about security, and as we'll see shortly, security certifications are thriving. Is it possible that Cisco and its certifications aren't as relevant as they used to be?

As we mentioned above, we also saw a drop in the relatively new certification for HashiCorp's Terraform (43%). That's a sharp decline —particularly since use of content about Terraform itself only declined 4.0%, showing that Terraform skills remain highly desirable regardless of the certification. A sudden drop in certification prep can be caused by a new exam, making older content out-of-date,

but that isn't the case here. Terraform certification certainly wasn't helped by HashiCorp's switch to a Business Source License or the subsequent fork of the Terraform project. IBM's pending acquisition of Terraform (set to close before the end of 2024) may have introduced more uncertainty. Is the decline in interest for Terraform certification an indicator of dissatisfaction in the Terraform community?
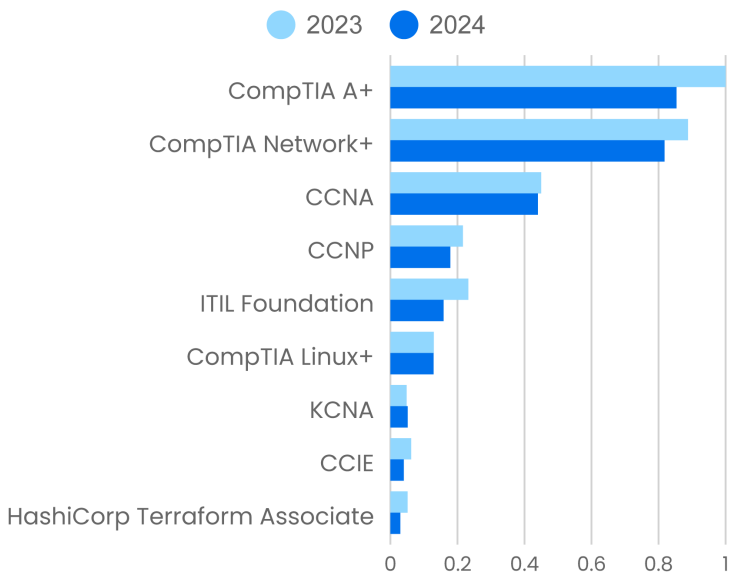
Figure 8. Certifications for IT

The Kubernetes and Cloud Native Associate (KCNA, up 6.3%) was a bright spot in IT certification. Whether or not Kubernetes is overly complex (perhaps *because* it's overly complex) and whether or not companies are moving out of the cloud, KCNA certification is a worthwhile asset. Cloud native applications aren't going away. And whether they're managing Kubernetes complexity by building developer platforms, using a Kubernetes provider, or using some other solution, companies will need people on their staff who can demonstrate that they have Kubernetes skills.

Cloud and cloud certifications

Content use for the major cloud providers and their certifications was down across all categories, with one exception: Use of content to prepare for Google Cloud certifications is up 2.2%.

What does that tell us, if anything? Are we looking at a "cloud repatriation" movement in full swing? Are our customers moving their operations back from the cloud to on-prem (or hosted) data centers? Last year, we said that we see very little evidence that repatriation is happening. This year? An article in *The New Stack* argues that cloud repatriation is gathering steam. While that might account for the decline in the use of cloud-related content, we still see little evidence that repatriation is actually happening. Two case studies (37signals and GEICO) don't make a trend. The ongoing expense of operating software in the cloud probably is greater than the cost of running it on-premises. But the cloud allows for scaling on demand, and that's important. It's true, few businesses have the sudden usage peaks that are driven by events like retail's Black Friday. But the cloud providers aren't just about sudden 10x or 100x bursts of traffic; they also allow you to scale smoothly from 1x to 1.5x to 2x to 3x, and so on. It saves you from arguing that you need additional infrastructure until the need becomes a crisis, at which point, you don't need to grow 1.5x; you need 5x. After moving operations to the cloud and experiencing a few years of growth—even if that growth is moderate—moving back to an on-premises data center will require significant capital expense. It will probably require gutting all the infrastructure that you haven't been using for the past year and replacing it with something up-to-date.

Does this mean that cloud providers are "roach motels," where you can move in but you can't move out? That's not entirely untrue. But the ease of scaling by allocating a few more servers and seeing a slightly higher bill the next month can't be ignored, even if those slightly higher bills sound like the proverbial story of boiling the frog. Evaluating vendors, waiting for delivery, installing hardware, configuring hardware, testing hardware—that's effort and expense that businesses are offloading to cloud vendors. The ability to scale fluidly is particularly important in the age of AI. Few companies have the skills needed to build on-premises infrastructure for AI, with its cooling and power requirements. That means either buying AI services directly from cloud providers or building infrastructure to host your own models. And of course, the cloud providers have

plenty of help for companies that need to use their high-end GPUs. (Seriously—if you want to host your AI application on-premises, see how long it will take to get delivery of NVIDIA's latest GPU.) The reality, as IDC concluded in a survey of cloud use, is that "workload repatriation from public cloud into dedicated environments goes hand in hand with workload migration to public cloud activities, reflecting organizations' continuous reassessment of IT environments best suited for serving their workloads." That is, there's a constant ebb and flow of workloads to and from public clouds as companies adapt their strategies to the business environment.
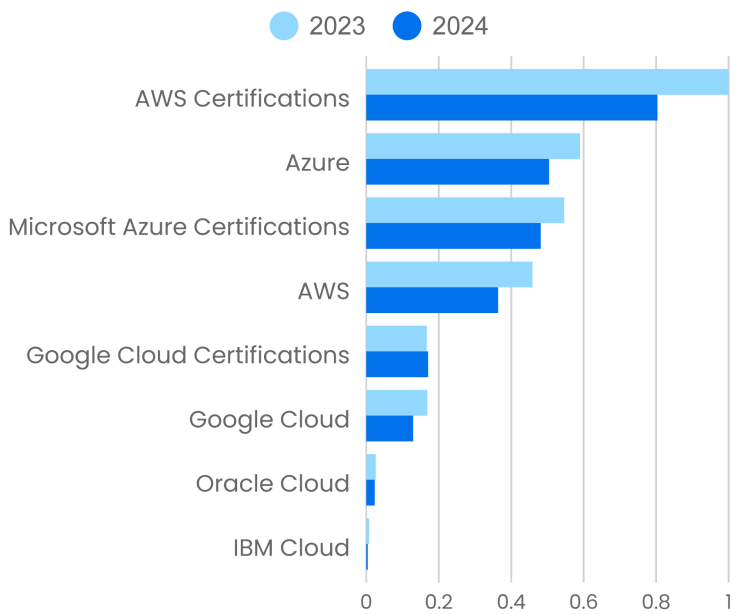
Figure 9. Cloud providers and certifications

The buzzword power of "the cloud" lasted longer than anyone could reasonably have expected, but it's dead now. However, that's just the buzzword. Companies may no longer be "moving to the cloud"; that move has already happened, and their staff no longer need to learn how to do it. Organizations now need to learn how to manage the investments they've made. They need to learn which workloads are most appropriate for the cloud and which are better run on-premises. IT still needs staff with cloud skills.

Security

Security Governance drove the most content use in 2024, growing 7.3% in the process and overtaking Network Security (down 12%). The rise of governance is an important sign: "Security" is no longer an ad hoc issue, fixing vulnerabilities in individual applications or specific services. That approach leads to endless firefighting and eventually failure—and those failures end up in the major news media and result in executives losing their jobs. Security is a company-wide issue that needs to be addressed in every part of the organization. Confirming the growing importance of security governance, interest in Governance, Risk, and Compliance (GRC) grew 44%, and Compliance grew 10%. Both are key parts of security governance. Security architecture also showed a small but significant increase (3.7%); designing a security architecture that works for an entire organization is an important part of looking at the overall security picture.

The use of content about Application Security also grew significantly (17%). That's a very general topic, and it perhaps doesn't say much except that our users are interested in securing their applications—which goes without saying. But what kinds of applications? All of them: web applications, cloud applications, business intelligence applications, everything. We get a bigger signal from the increase in Zero Trust (13%), a particularly important strategy for securing services in which every user, human or otherwise, must authenticate itself to every service that it uses. In addition, users must have appropriate privileges to do what they need to do, and no more. It's particularly important that zero trust extends authentication to non-human users (other computers and other services, whether internal or external). It's a response to the "hard, crunchy outside, but soft chewy inside" security that dominated the 1990s and early 2000s. Zero trust assumes that attackers can get through firewalls, that they can guess passwords, and that they can compromise phones and computers when they're outside the firewall. Firewalls, good passwords, and multifactor authentication systems are all important— they're the hard, crunchy outside that prevents an attacker from getting in. Zero trust helps keep attackers outside, of course—but more than that, it limits the damage they can do once they're inside.
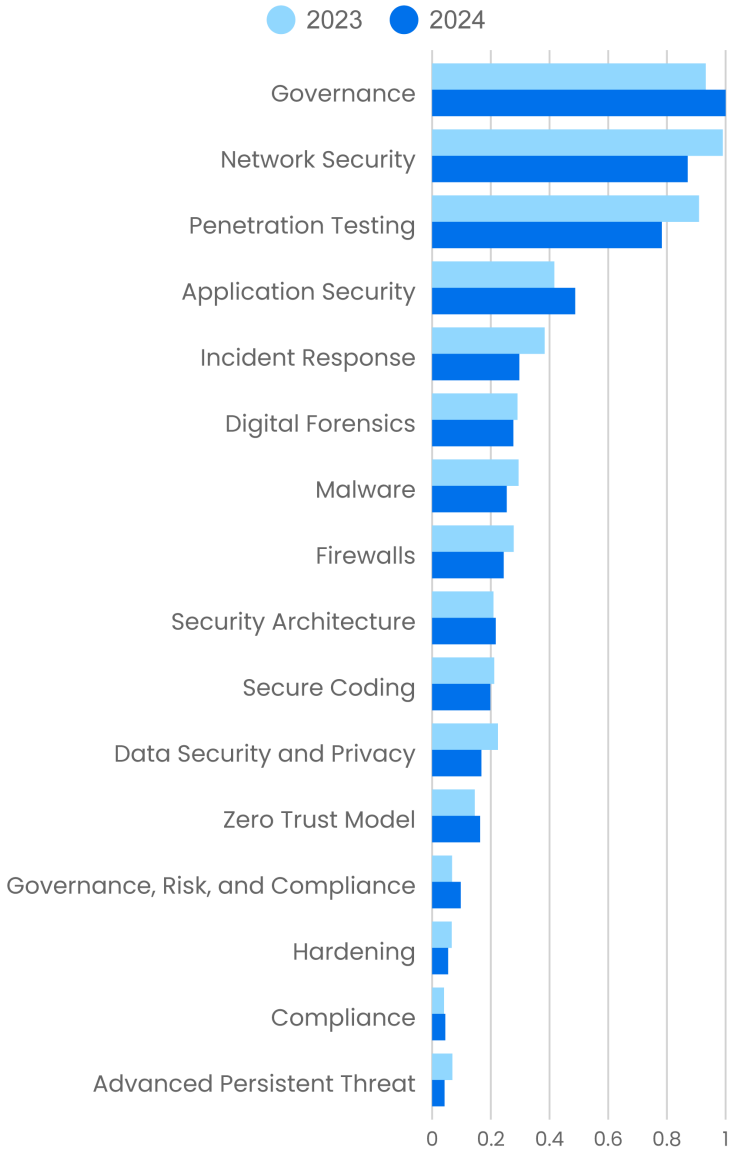
Figure 10. Security skills

We're puzzled by the drop in use of content about Network Security, which corresponds roughly to the drop in Cisco certifications. Network Security is still the second most widely used skill, but it's down 12% from 2023 to 2024. Perhaps network security isn't deemed

as important when employees wander in and out of company networks and applications are distributed between in-house servers and the cloud. We hope that our users aren't making that mistake. A bigger issue is that networks haven't changed much in the past few years: We're still using IPv4; we're still using routers, switches, and firewalls, none of which have changed significantly in recent years. What has changed is the way security is implemented. Cloud computing and zero trust have moved the focus from big-iron networking devices to interactions between systems, regardless of how they are connected.

Security certifications

Security certification has been one of the biggest growth areas on our platform. As I've said elsewhere, security professionals love their certifications. There's a good reason for that. In most other specialties, it's possible to build a portfolio of programs you wrote, systems you architected, sites you've designed. What can a security person say in a job interview? "I stopped 10,000 people from logging in last year?" If you've ever monitored a public-facing Linux system, you know that claim means little. Security is cursed with the problem that the best news is no news: "Nothing bad happened" doesn't play well with management or future employers. Neither does "I kept all the software patched, and spent time reading CVEs to learn about new vulnerabilities"—even though that's an excellent demonstration of competence. Certification is a way of proving that you have certain skills and that you've met some widely recognized standards.

The CISSP (up 11%) and CompTIA Security+ (up 13%) certifications are always at the top of our lists, and this year is no exception. Our *State of Security in 2024* report showed that CISSP was the certification most commonly required by employers. If there's a gold standard for security skills, CISSP is it: It's a thorough, comprehensive exam for people with more than five years of experience. CompTIA Security+ certification has always trailed CISSP slightly in our surveys and in platform performance, but its position in second place is uncontested. Security+ is an entry-level certification; it's particularly desirable for people who are starting their security careers.

Security certification was especially important for government users. For most industry sectors, usage focused on programming skills in Java or Python, followed by artificial intelligence. The government sector was a strong outlier. Security and IT certifications were by far the most important topics. CompTIA Security+ and CISSP (in that order) led.

Moving beyond CISSP and Security+, many of the other security certifications also showed gains. Certified Ethical Hacker (CEH) was up 1.4%, as was the less popular CompTIA PenTest+ certification (3.3%). Certified Cloud Security Professional was up 2.4%, somewhat less than we'd expect, given the importance of the cloud to modern IT, but it's still a gain. ISACA's Certified in Risk and Information Systems Control (CRISC) was up 45%, Certified Information Security Manager (CISM) grew 9.3%, and Certified Information Security Auditor (CISA) was up 8.8%; these three certifications are strongly associated with security governance. The most significant declines were for the CompTIA Cybersecurity Analyst (CySA+) certification (down 13%) and CCNA Security (down 55%). The drop in CCNA Security is extreme, but it isn't unexpected given that none of the Cisco certifications showed an increase this year.

We're missing one important piece of the security certification puzzle. There's no data on AI security certifications—and that's because there aren't any. Software that incorporates AI must be built and operated securely. That will require security experts with AI expertise (and who can demonstrate that expertise via certifications). We expect (or maybe a better phrase is "we hope") that lack will be addressed in the coming year.
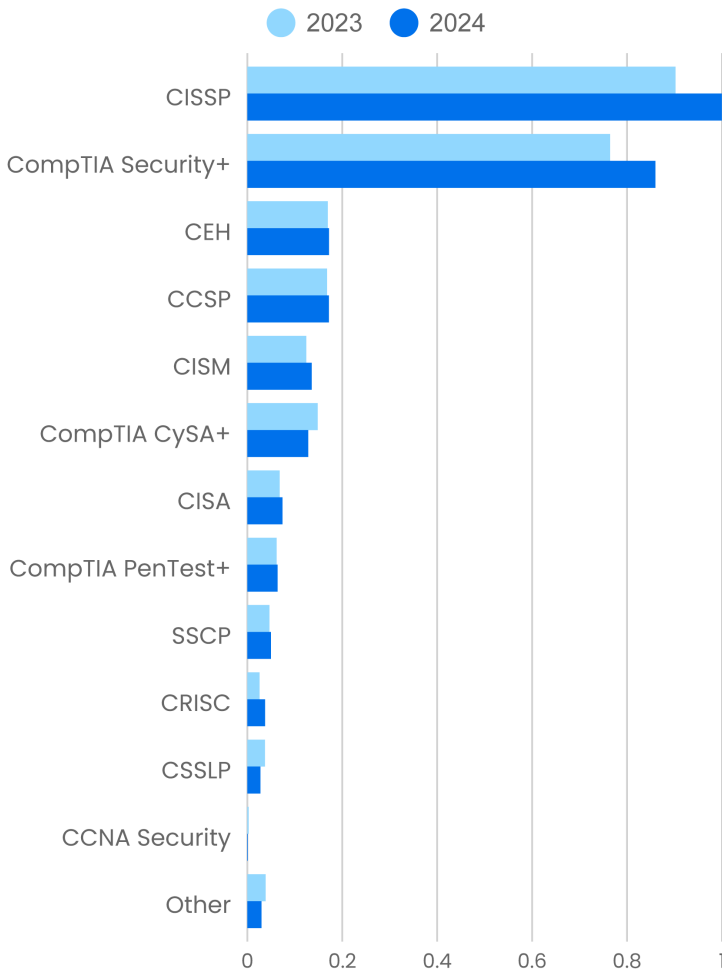
Figure 11. Security certifications

Professional Development

Professional development continues to be an important growth area for our audience. The most important skill, Professional Communication, grew 4.5%—not much but significant. We saw a 9.6% increase in users wanting to know more about Engineering Leadership, and a 21.5% increase in users using content about Personal Productivity.

Project Management was almost unchanged from 2023 to 2024 (up 0.01%), while the use of content about the Project Management Professional (PMP) certification grew 15%. Interest in Product Management declined 11%; it seems to be a skill that our users are less interested in. Why? For the past few years, product manager has seemed to be a trendy new job title. But in last year's report, Product Management only showed a small gain from 2022 to 2023. Is interest in Product Management as a skill or as a job title fading?

Figure 12. Professional development and skills

We also saw a 7.9% decline in Leadership (aside from Engineering Leadership), and a huge 35% decline for IT Management. Are we to blame these on the corporate layoff cycle? That's possible, but it's too easy. IT may be affected by a general trend toward simplification and platform engineering, as we've discussed: A platform engineering group can do a lot to reduce cognitive overhead for developers, but it also reduces the need for IT staff. A platform engineering group doesn't have to be large; is the need for IT staff shrinking?

The decline in Leadership may be because it's a vague, nonspecific term, unlike Engineering Leadership (which is up). Engineering Leadership is concrete and it's something our engineering-oriented audience understands.

New Initiatives

In 2024, we introduced several new features on the O'Reilly learning platform, including badges, quizzes, and a new version of O'Reilly Answers. What are they telling us?

Badges and Quizzes

We started a badging program late in 2023: Users from business accounts can earn badges for taking courses and completing quizzes. We won't go into the program details here, but since the program started, users have earned nearly 160,000 badges. We're still building the program, but we're encouraged by its first year.

Badges can give us more insight into what our users are learning. The most popular badges are for Python skills, followed by GPT and prompt engineering. Generative AI and machine learning are also high on the list. Kubernetes, despite its decline in units viewed, was the fourth-most-frequently-acquired badge, with almost the same number of badges earned as software architecture. Linux, SQL, professional communication, and Java rounded out the top 11. (Yes, 11—we wanted to include Java). The difference between Java and Python is striking, given that the use of content about these skills is similar. (Python leads Java, but not by much.) Oracle has a highly regarded Java certification program, and there's really no equivalent for Python. Perhaps our users recognize that obtaining a Java badge is superfluous, while obtaining badges for Pythonic skills is meaningful?

Quizzes are closely tied to badges: If a final quiz exists for a course or for a book, students must pass it to earn their badge. Quiz usage appears to follow the same trends as badging, though it's premature to draw any conclusions. While a few legacy quizzes have been on the platform for a long time (and aren't connected to badging), the push to develop quizzes as part of the badging program only began in June 2024, and quiz usage is still as much a consequence of the time the quiz has been available on the platform as it is of the skill for which it's testing.
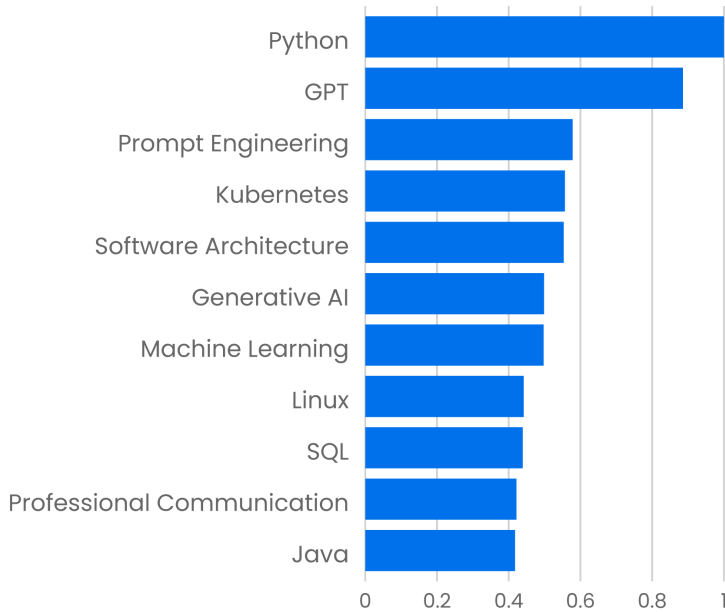
Figure 13. Top badges earned (relative to Python)

We can also look at the expertise required by the badges that were earned. All of our content is tagged with a skill level: beginner, beginner-intermediate, intermediate, intermediate-advanced, or advanced. 42% of the badges were earned for content judged to be intermediate. 33% of the badges were earned for beginner content, while only 4.4% were for advanced content. It's somewhat surprising that most of the badges were earned for intermediate-level content, though perhaps that makes sense given the badge program's B2B context: For the most part, our users are professionals rather than beginners.
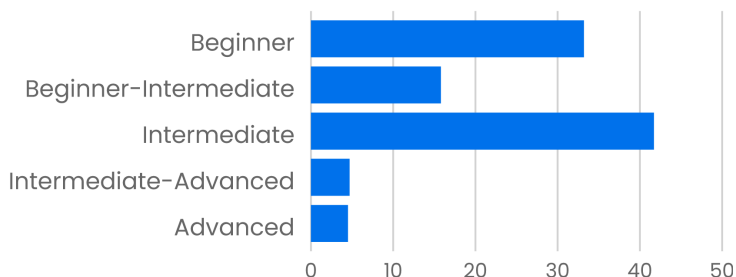
Figure 14. Badges earned by expertise level (percent)

Answers

One of our most important new features in 2024 was an upgrade to O'Reilly Answers. Answers is a generative AI-powered tool that allows users to enter natural language questions and generates responses from content in our platform. Unlike most other generative AI products, Answers always provides links to the original sources its responses are based on. These citations are tracked and used to calculate author royalties and payments to publishing partners.

So the obvious question is: What are our users asking? One might guess that the questions in Answers would be similar to the search terms used on the platform. (At this point, Answers and search are distinct from each other.) That guess is partly right—and partly wrong. There are some obvious differences. Common search terms include book titles, author names, and even ISBNs; titles and author names rarely appear in Answers. The most common searches are for single words, such as "Python" or "Java." (The average length of the top 5,000 searches in September 2024 was two words, for instance.) There are few single word questions in Answers (though there are some); most questions are well-formed sentences like "How many ways can you create a string object in Java?" (The average question length was nine words.)

To analyze the questions from O'Reilly Answers, we essentially turned them back into single-word questions. First, we eliminated questions from a "question bank" that we created to prime the pump, as it were: Rather than requiring users to write a new question, we offered a list of prewritten queries they could click on. While there's undoubtedly some useful signal in how the question bank was used, we were more interested in what users asked of their

own volition. From the user-written questions, we created a big "bag of words," sorted them by frequency, and eliminated stopwords. We included a lot of stopwords that aren't in most lists: words like "data" (what does that mean by itself?) and "chapter" (yes, you can ask about a chapter in a book, but that doesn't tell us much).

With that background in mind, what were the most common words in Answers and in searches? In order:

Answers	Search queries
Python	Python
Java	Machine learning
Management	Kubernetes
Key	Java
Model	Rust
Security	React
File	AWS
Architecture	CISSP
AI	C++
System	Linux
Service	Docker
Project	SQL
Learning	JavaScript

There's an obvious difference between these two lists. The Answers list consists mostly of words that could be part of longer questions. The Search list is made up of topics and skills about which one might want information. That's hardly surprising or insightful. We've said most searches on the platform are single-word searches, which means that those words have to be stand-alone skills or topics, like Python or Java. Likewise, Answers was built to allow users to ask more detailed, in-depth questions and get focused answers from the content on our platform—so rather than seeing single word searches, we're seeing common words from longer questions. Maybe that's a self-fulfilling prophecy, but it's also showing that Answers is working the way we intended.

There's a little more signal here. Python and Java are the two top programming languages on both lists, but if we look at search queries, machine learning and Kubernetes are sandwiched between the two languages. That may just be a result of our users' experiences

with services like ChatGPT. Programmers quickly learned that they can get reasonable answers to questions about Java and Python, and the prompts don't have to be very complex. My personal favorite is "How do you flatten a list of lists in Python?," which can be answered by most chatbots correctly but isn't meaningful to our search engine.

Kubernetes raises a different question: Why is it the third-most-common search engine query but doesn't appear among the top words on Answers? (It's the 90th-most-common word on Answers, though the actual rank isn't meaningful.) While Kubernetes is a topic that's amenable to precise questions, it's a complex tool, and coming up with precise prompts is difficult; writing a good question probably requires a good understanding of your IT infrastructure. You might need to understand how to solve your problem before you can ask a good question about how to solve your problem. A search engine doesn't face problems like this. It doesn't need additional information to return a list of resources.

Then what about words like Rust and Linux, which are high on the list of common searches, but not in the top 13 for Answers? It's relatively easy to come up with specific questions about either of these—or, for that matter, about SQL, AWS, or React. SQL, AWS, and Linux are reasonably close to the top of the Answers word list. If we just concern ourselves with the order in which words appear, things start to fall into place: AWS (and cloud) follow learning; they are followed by Linux, followed by SQL. We're not surprised that there are few questions about CISSP on Answers; it's a certification exam, so users are more likely to want test prep material than to ask specific questions. Rust and React are still outliers, though; it's easy to ask precise and specific questions about either of them. Rust is still unfamiliar to many of our users—could the explanation be that our customers want to learn Rust as a whole rather than ask specific questions that might only occur to someone who's already learned the language? But if you accept that, React still remains an outlier. We may know the answers next year, at which time we'll have a much longer track record with Answers.

The Coming Year

That wraps up last year. What will we see this year? We've given hints throughout this report. Let's bring it all together.

AI dominated the news for 2024. It will continue to do so in 2025, despite some disillusionment. For the most part, those who are disillusioned aren't the people making decisions about what products to build. While concern about jobs is understandable in a year that's seen significant layoffs, we don't believe that AI is "coming for your job." However, we do believe that the future will belong to those who learn how to use AI effectively—and that AI will have a profound impact on every profession, not just IT and not just "knowledge workers." Using AI effectively isn't just about coming up with clever prompts so you can copy and paste an answer. If all you can do is prompt, copy, and paste, you're about to become superfluous. You need to figure out how to work with AI to create something that's better than what the AI could do by itself. Training employees to use AI effectively is one of the best things a company can do to prepare for an AI-driven future. Companies that don't invest in training will inevitably fall behind.

In the coming year, will companies build AI applications on top of the giant foundation models like GPT-4, Claude, and Gemini? Or will they build on top of smaller open models, many of which are based on Meta's Llama? And in the latter case, will they run the models on-premises (which includes the use of hosting and colocation providers), or will they rent use of these open AI models as a service from various providers? In the coming year, watch carefully what happens with the small open models. They already deliver performance almost as good as the foundation models and will undoubtedly be the basis for many AI applications. And we suspect that most companies will run these models in the cloud.

Security is the other significant growth area. Companies are waking up to the need to secure their data before their reputations—and their bottom lines—are compromised. Waking up has been a long, slow process that has sunk the careers of many CEOs and CIOs, but it's happening. Our users are studying to gain security certifications. We see companies investing in governance and putting in company-wide policies to maintain security. In this respect, AI cuts both ways. It's both a tool and a danger. It's a tool because security professionals need to watch over huge streams of data, looking for the anomalies

that signal an attack; it's a tool because AI can digest sources of information about new threats and vulnerabilities; it's a tool because AI can automate routine tasks like report generation. But it's also a danger. AI-enabled applications increase an organization's threat surface by introducing new vulnerabilities, like prompt injection, that we're only now learning how to mitigate. We haven't yet seen a high-profile attack against AI that compromised an organization's ability to do business, but that will certainly happen eventually—maybe in 2025.

Whatever happens this year, AI will be at the center. Everyone will need to learn how to use AI effectively. AI will inevitably reshape all of our professions, but we don't yet know how; we're only starting to get glimpses. Is that exciting or terrifying? Both.

About the Author

Mike Loukides is vice president of content strategy for O'Reilly Media, Inc. He's edited many highly regarded books on technical subjects that don't involve Windows programming. He's particularly interested in programming languages, Unix and what passes for Unix these days, and system and network administration. Mike is the author of *System Performance Tuning* and a coauthor of *Unix Power Tools*. Most recently, he's been fooling around with data and data analysis, exploring languages like R, Mathematica, and Octave, and thinking about how to make books social. Mike can be reached on X as @mikeloukides and on LinkedIn.
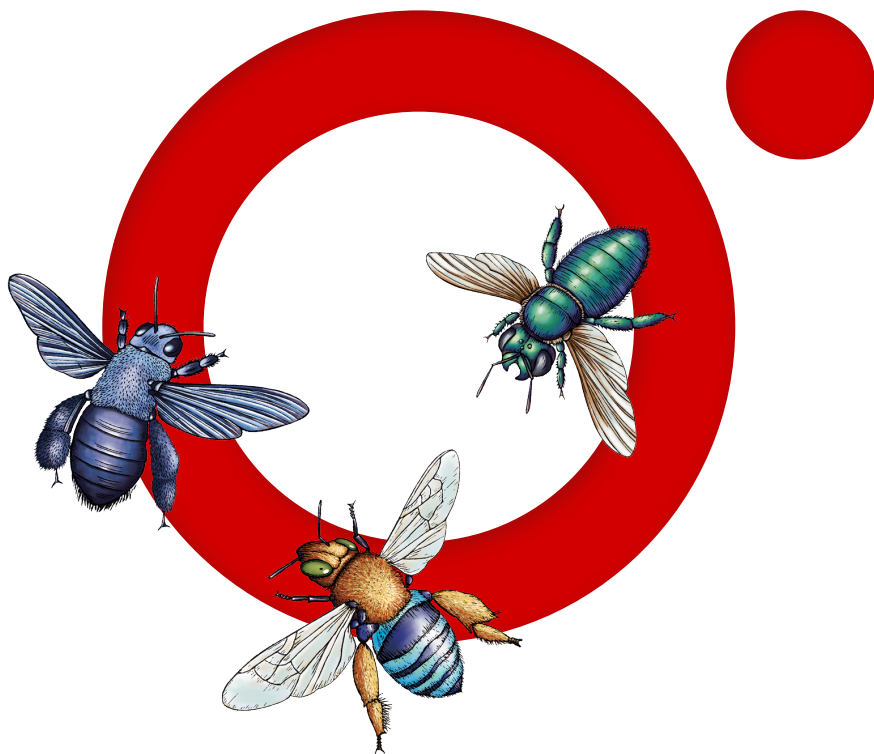
O'REILLY®

Learn from experts.
Become one yourself.

60,000+ titles | Live events with experts
Role-based courses | Interactive learning
Certification preparation

**Try the O'Reilly learning platform
free for 10 days.**